

Ares Acoustical System

Modeler Module

User Manual

Acoustical and Audio Measurement and Design Tools

McIntosh Applied Engineering, LLC (MAE)

MAELLC.COM

info@MAELLC.COM

support@MAELLC.COM

Copyright 2012-2015 by McIntosh Applied Engineering, LLC

Eden Prairie, MN

USA

Notice

Ares is copyrighted and licensed by McIntosh Applied Engineering, LLC (MAE). Ares and all of its hardware and software components are provided "as is". MAE makes no representations or warranties concerning the compatibility of Ares to the user's computer system, or any potential damage caused to any computer, peripherals, digital storage systems, or personal physical safety. Further, the accuracies of the measurement, modeling and design components are not warranted and should not be used as the sole source of evaluating an engineered system or component for commercial suitability or physical safety.

The user agrees not to distribute the Ares software to any non-licensed third party, attempt to disable the licensing system, or reverse engineer the operation of the program or its hardware components.

Table of Contents

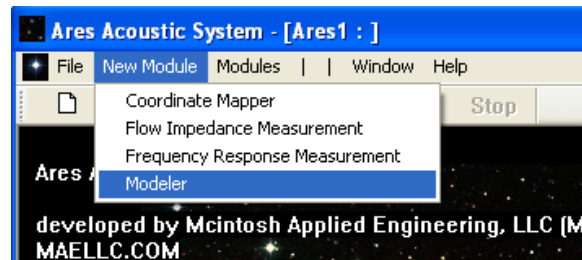
Notice.....	2
Table of Contents.....	2
Document Revision History.....	3
Acoustic Modeler Module	4
Interface layout	4
Building a model.....	5
Solution Errors.....	9
Displaying Model Results.....	10
Duplicating, deleting and copying elements.....	14
Duplicating Models.....	15
Options Menu.....	18
Node types	18
Disabling model sections	21
Modifying Element Parameters	22
Iterating a Parameter	24
Variables	26
Global Parameters	28
SubCircuits.....	29
End Corrections	32
Graph Element	33
Acoustic Material Database	37
Mechanical Modeling	42
Modeling Elements	45
Electrical Elements	46
Mechanical Elements	56
Acoustical Elements.....	61
Thermal Elements	83
Examples.....	85
Example 1 Bass Reflex Speakerphone	85
Example 2: Microphone directivity	90
Example 3: Rigid Rectangle Model of Microphone on Tablet.....	93

Document Revision History

Version	Date	Comments
1.00	October, 2012	Initial documentation release
1.01	March 2015	Separated module manuals
1.02	April 2015	Added electrical-acoustical transfer function. Added R2 and L2 to speaker element. Added the following elements to the Modeler: electrical-acoustical transfer function ,differential voltage amplifier, current amplifier, and transressitance amplifier. Added the ability to inverse transfer functions imported into the Modeler.
1.03	July 2015	Added porous material to fluid properties, subcircuits, and the ability to import Klippel *.KLARES speaker parameter files.
1.04	November 2015	Added the following elements: Knowles and Sonion receiver and omni microphone elements, electrical-to-acoustical transfer function, diode, N and P channel MOSFETs, DC voltage and current sources. Also added a DC solver when a nonlinear electrical device is present in the model (such as the diode and MOSFETs). Modified ear simulator element to fix IEC711 correction and changed sensing node from freefield to diffuse field.

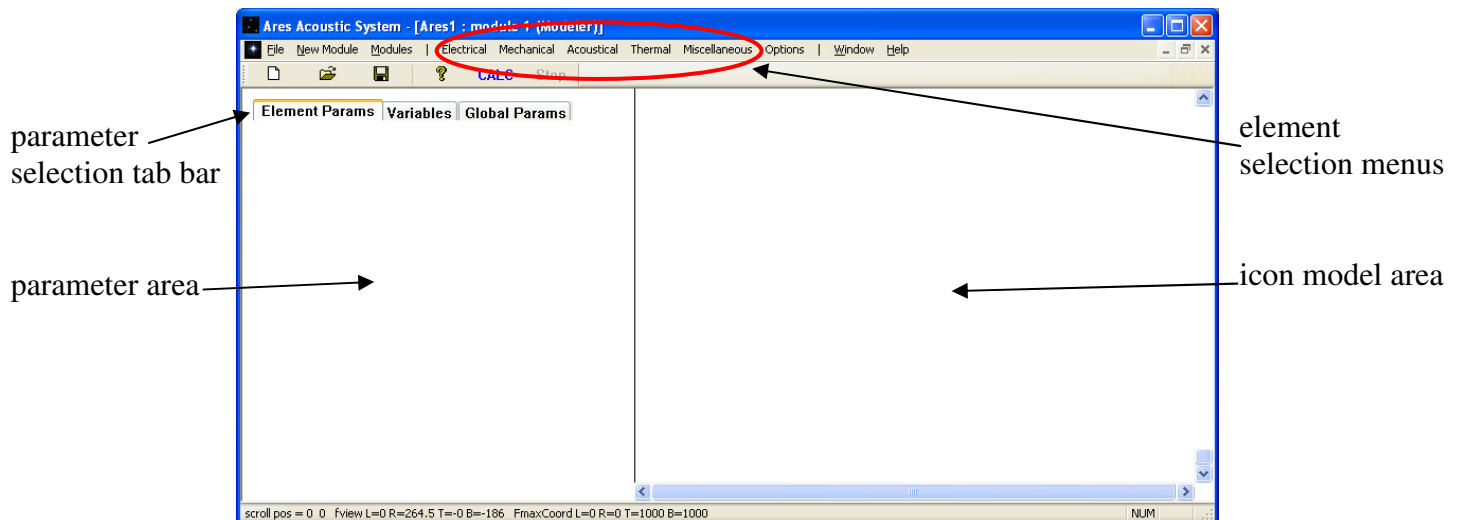
Acoustic Modeler Module

The Ares Modeler performs a linear, frequency domain lumped parameter analysis of electrical, mechanical, acoustical, and thermal systems. The main focus of the model is acoustical modeling and acoustical components. To launch the modeler, open the New Modeler menu and select Modeler as shown below.



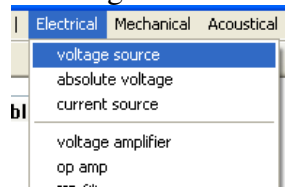
Interface layout

When a new Modeler module is created, the interface appears as shown below. Note that a new set of menu entries has appeared between the two pipe || bars. These menus are grouped into the four element types (electrical, mechanical, acoustical, thermal) and a miscellaneous menu. The Miscellaneous menu contains the graph element, which is required to show modeling results.

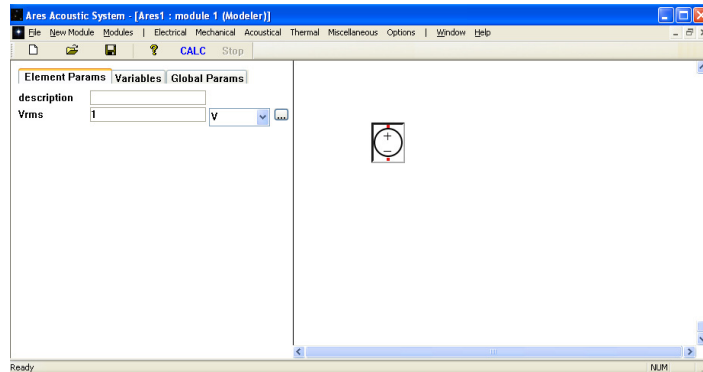


Building a model

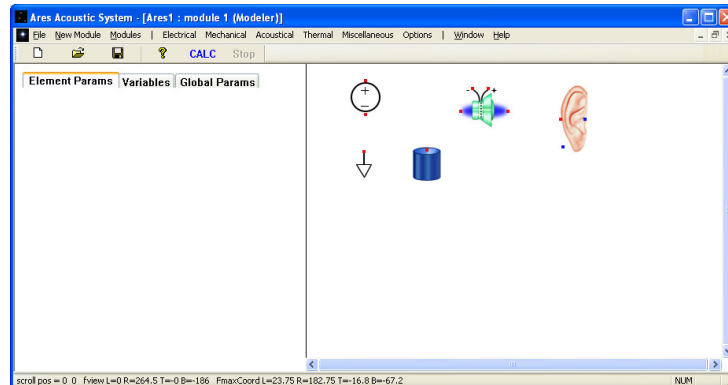
To introduce the workings of the modeler, we'll jump right in and build a model. In the Electrical menu, select a voltage source



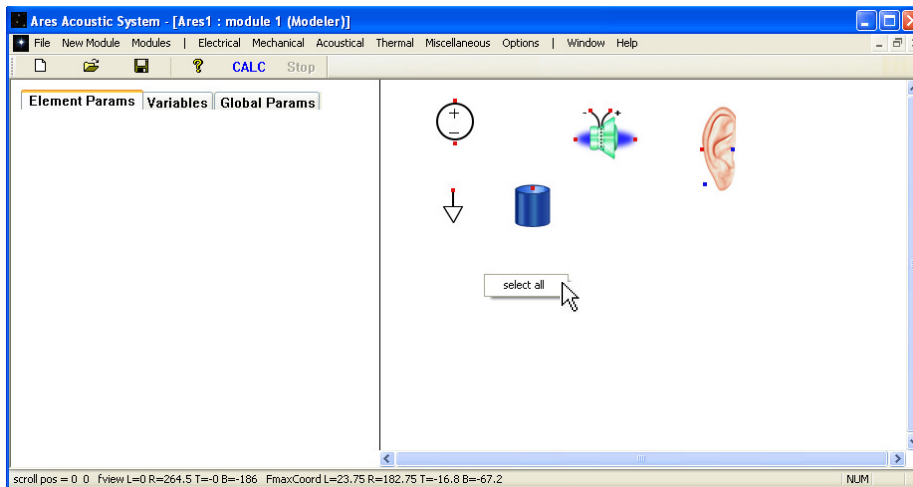
and drag the voltage source icon that's created to a position in the upper left corner of the icon model area of the Ares window. Click the left mouse button to "drop" the icon. Note that you can left-click + drag the icon again to move it to another position.



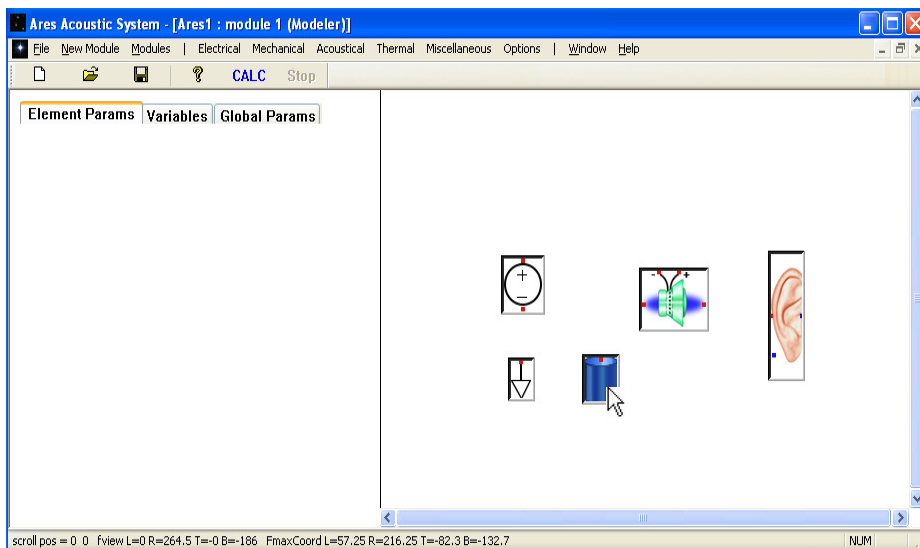
Do the same with the following icons: Electrical/ground, Acoustical/speaker, Acoustical/volume, Acoustical/ear simulator. Try to create the model shown below.



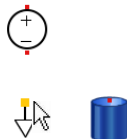
Note that you can move all of the elements by right clicking in the icon model area, and then choosing “select all” from the pop-up menu.



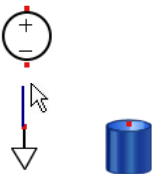
Then left click on any of the selected icons and drag the icons around to a better position in the icon model area.



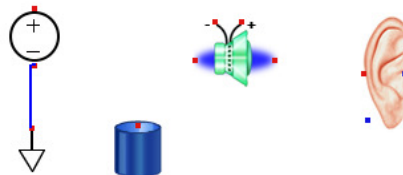
Next, the icons have to be connected together. To do this, first click the mouse anywhere in the icon model area, but not on an element. This will deselect all of the elements. Next, move the mouse over the red node on the ground element.



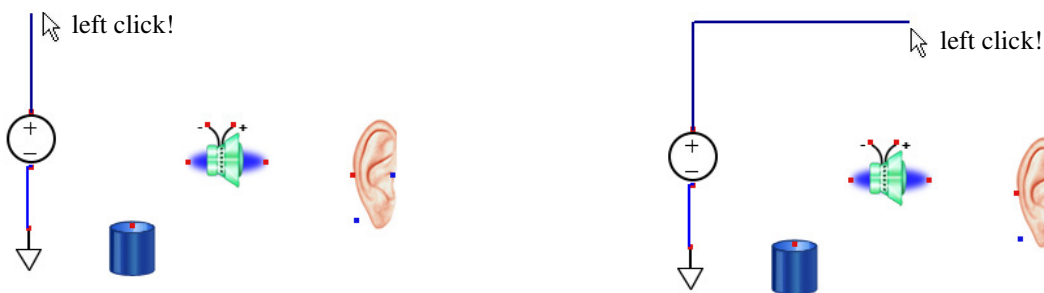
It will turn yellow, indicating that you're over that connector node. Then left click on the now yellow node to begin the creation of a blue connector line.



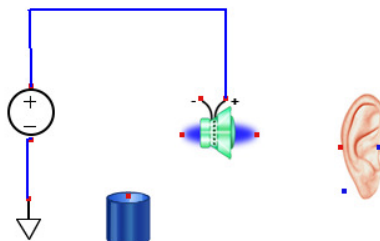
Next, move the mouse over the lower node on the voltage source and left click again. This connects the ground element to the lower voltage terminal of the voltage source.



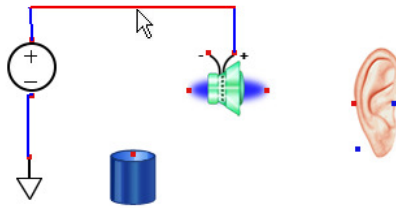
You can manually route a connector by left clicking along a path between the two element nodes. For example, connect the + voltage node to the + speaker node by clicking twice to bring the connector above both elements.



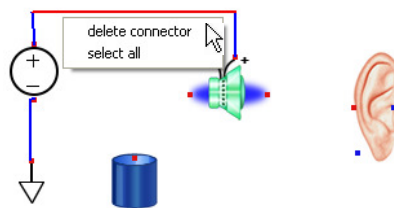
Then finally click on the + speaker terminal to complete the connection. Ares will automatically align the last click point to be in line with the last node you click on.



Note that one section of a connector can be moved by moving the mouse over it and dragging.

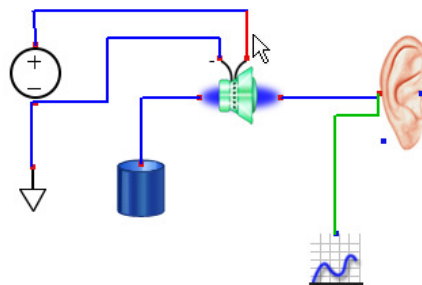


A connector can be deleted by right clicking on it and selecting delete connector. But let's not delete the connector we just created.

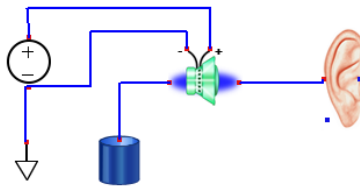


Also, once you start a connector, you can cancel it by pressing the "Esc" (escape) key. (Another convenient way to delete a connector is to click on it close to an icon for editing, and then just press the "Esc" (escape) key to cancel it.)

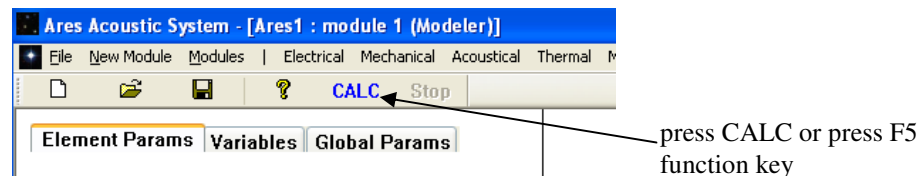
Once a connector has been put in place, it can be detached from a node by clicking on the last line segment of that connector. If you want to disconnect the connector from the + terminal of the speaker, you'd left click on the line segment connected to the + node as shown below. This will detach the connector from the speaker, but keep it connected to the voltage source element. You can then re-route the connector, or simply press the "Esc" (escape) button to delete the connector altogether. This ability to move a connector is especially useful for graph connectors when you want to plot the data at different nodes. When making a connector, you may want to make sure that the last line segment has a significant length to make it easier to click on should you want to move the connector.



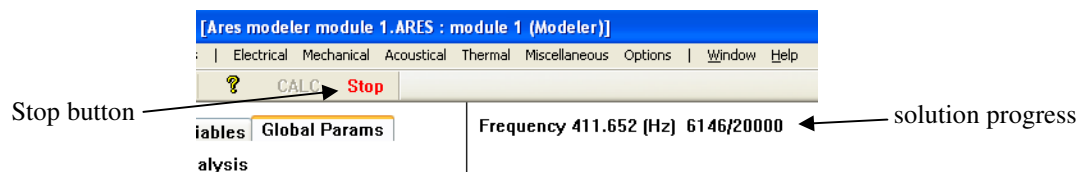
Finish connecting the model as shown below.



To calculate the solution, either press the CALC button in the tool bar, or press the F5 key.



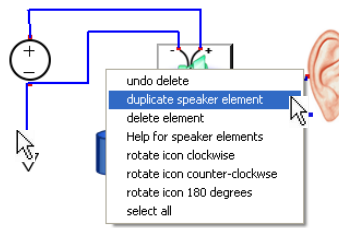
The calculation progress status will be displayed in the upper left corner of the icon model area. In this case, the model is so simple, it will likely finish in a fraction of a second, so the solution state will just flash up onto the screen. For more complicated models, or more frequency points than the default 200, the calculation will take longer. If the solution is taking too long, the Stop button to the right of the CALC button can be pressed to terminate the solution and fewer frequency points can be entered to speed up the solution. For this simple model, the number of frequency steps has been increased to 20000 so the progress could be captured. This is shown below. Note that the Stop tool bar button is enabled.



Solution Errors

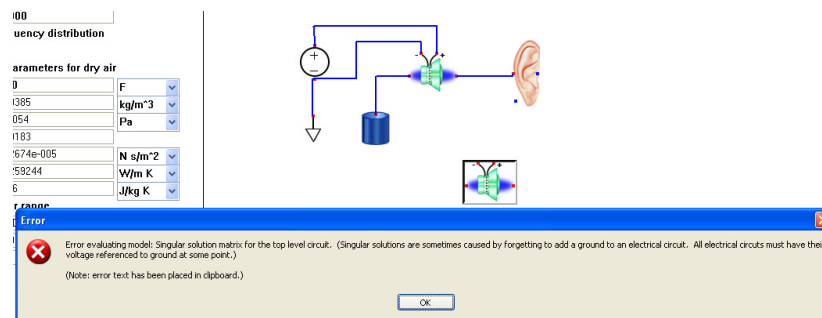
Occasionally you may find that the solver generates an error. This is usually caused by an unrealizable solution. For instance, if you connect a 1V source to ground, Ares won't be able to find a solution. A more common problem is to have an unconnected element in the model. For instance, if we add another volume element, but we don't connect it to the rest of the circuit, the pressure in the volume will be undefined and an error will result.

If you want to create an element and one already exists in your model, a convenient way to produce a new one is simply to right click on the existing element and select *duplicate volume element* from the pop-up menu. We'll do that for the speaker element.

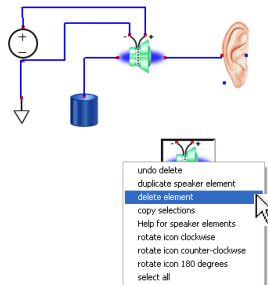


A big advantage of duplicating elements instead of creating new ones from the element menus is that the duplicated elements will have the same parameter values as the element you duplicated. The elements from the menus will have default values, which probably won't be appropriate for your model. However, we haven't gotten to how to change the elements' parameters yet.

Once the speaker element is duplicated, pressing F5 will generate a solution error because the new speaker element has too many undriven nodes and a unique solution couldn't be found.

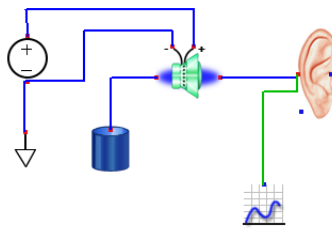


Correct the model by right clicking on the new speaker, and this time select *delete element*. Then press F5 again to recalculate the model.

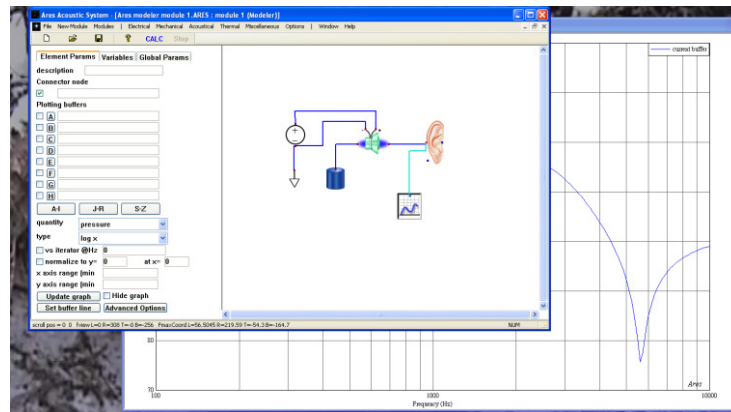


Displaying Model Results

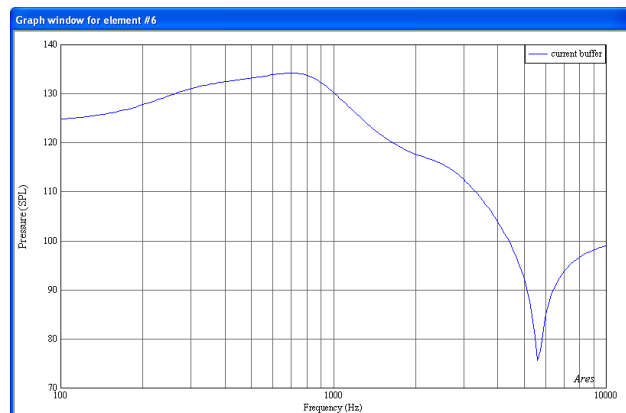
Once the model solution has been calculated, each of the nodes will contain solution information. For electrical nodes this will be voltage and current, for mechanical this will be force and velocity, and for acoustical this will be pressure and volume velocity. To plot these quantities, a graph element must be created and connected to the node whose values you wish to display. The graph element is in the *Miscellaneous* menu. Place the graph element as shown below and connect to the artificial ear input node.



When a graph element is used, a new window is created. It will appear somewhere on the desktop, completely separate from the main Ares window. Shown below, the graph window is directly behind the main Ares window. You'll need to position and size the Ares and graph windows to make them both visible and accessible as needed.



The graph window that was created is shown below. By default, Ares plots the pressure in SPL. However there are many more graphing options.



You can create as many graph elements as you like, with each generating their own graph window. This way, you can see the solution at different nodes in the model.

By clicking on the graph icon and selecting the *Element Params* tab, the graph options are displayed. These are shown below on the left. By pressing the *Advanced Options* button, the dialog box on the right appears. These constitute the graphing options available in the modeler. To get the most out of the modeler, you will need to familiarize yourself with all of these options. The options are extensive and will be discussed in their own sections later on. For now, we'll just make use of the plotting buffers to show how model results can be saved.

Element Params

Variables

Global Params

description

Connector node

☒

Plotting buffers

☐ A
☐ B
☐ C
☐ D
☐ E
☐ F
☐ G
☐ H

A-I

J-R

S-Z

quantity

pressure

type

log x

☐ vs iterator @Hz

0

☐ normalize to y=

0

at x=

0

x axis range (min)

y axis range (min)

Update graph

Hide graph

Set buffer line

Advanced Options

Graph options

File Edit

Quantity to plot

☒ magnitude
☐ real & imaginary
☐ phase
☐ power
☐ leak tolerance
☐ time response

Magnitude format

☐ linear
☒ dB narrow band
☐ dB full octave
☐ dB 1/3rd octave
☒ RMS level
☐ peak level

Legend options

☐ show buffer letter
☒ auto-position
☐ fixed position
☐ hide

Metric to display in legend

☐ average level
☐ 300-3000 Hz average level
☐ P50 weighted average level
☐ P50 wghd 300-3000 Hz average level
☐ Zwicker loudness
☐ P50 weighted Zwicker loudness
☐ RLR narrow band (176 to 4590 Hz)
☐ RLR wide band (88 to 9000 Hz)
☐ SLR narrow band (176 to 4590 Hz)
☐ SLR wide band (88 to 9000 Hz)

Shift buffer data by dB value

buffer

A

dB

0

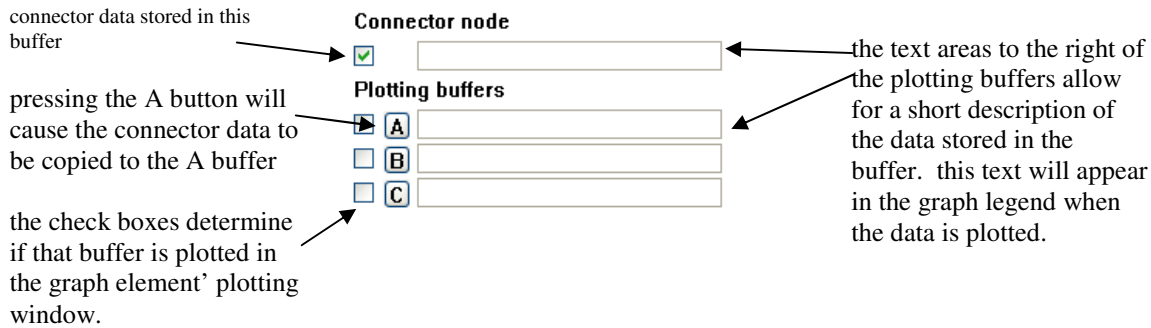
apply

Buffer weightings

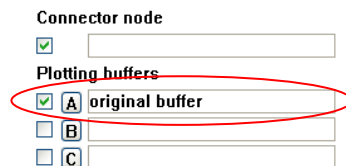
current	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
A weighting	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
B weighting	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
C weighting	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
P50 weighting	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
psophometric weighting	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
plot using negative kinetic value	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

OK

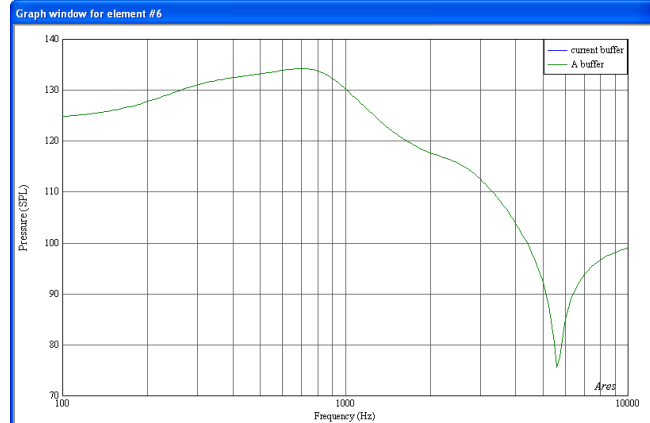
Suppose you would like to see how the speaker output to the ear simulator changes as the rear volume element size is decreased. The graph element uses “buffers” to hold the node data. The buffer labeled “Connector node” holds the data for the node that the graph connector is connected to. In this case, it’s the ear simulator node. By pressing one of the Plotting buffer buttons (A-Z), the Connector node data is copied to a buffer.



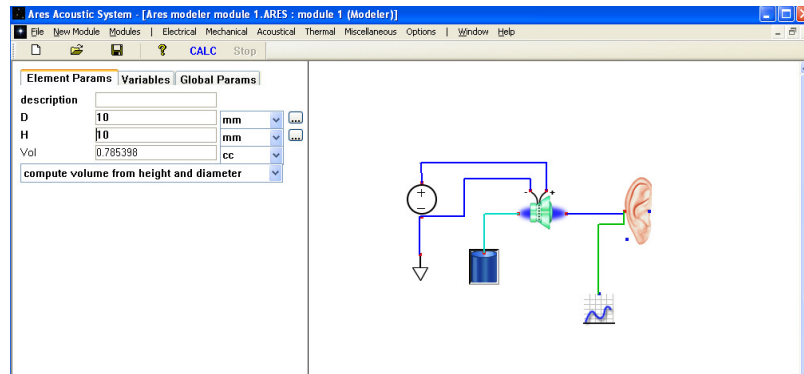
For this example, press the A button and type “original volume” into the A buffer description, and then check the A buffer box as shown below.



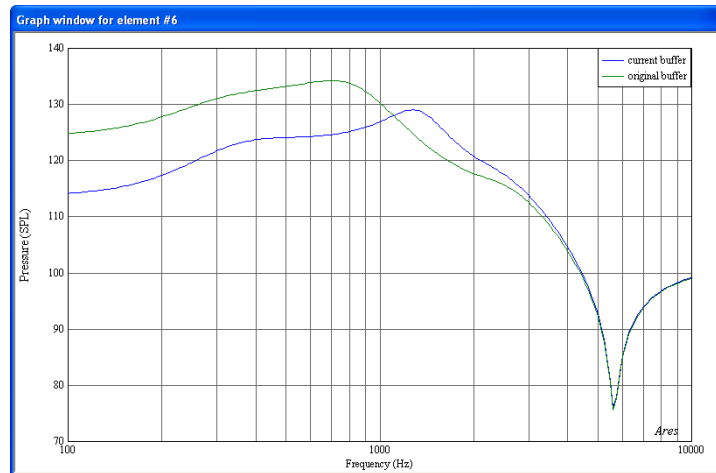
This causes a second graph to appear in the graph window, but since the data in buffer A is the same as that in the Connector node buffer, the two curves lie exactly on top of each other.



Now left click on the volume element so its parameters appear in the Element Params tab. Change the volume's height H to 10 mm and press the CALC button.

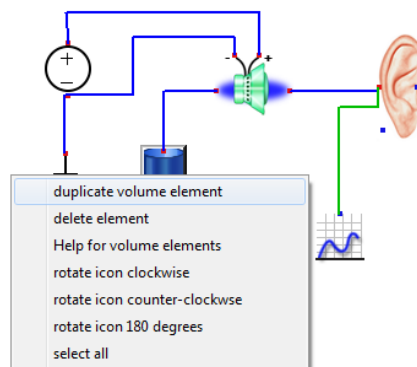


The graph now shows the results of the new calculation in blue, and the A buffer with the original volume size in green. It can clearly be seen that the effects of decreasing the rear volume reduced the low frequency response, and shifted the main speaker resonance to just above 1 kHz.

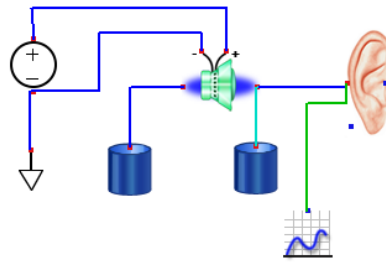


Duplicating, deleting and copying elements

If we wish to add an additional volume element between the speaker and the ear simulator, rather than going to the *Acoustical* menu and selecting *volume*, you can simply right click on an existing volume element and select *duplicate volume element* from the pop-up menu.



You can now drag the new volume element with the mouse and place it just below the speaker. Then connect it to the speaker with a connector.

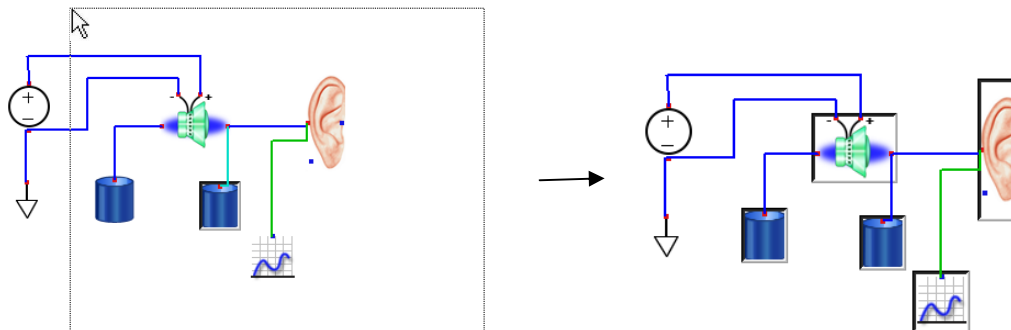


Pressing the F5 key or the CALC button will cause the model to be recomputed.

One advantage of using the right click feature to duplicate an element is that the new element will have the same parameter values.

You can delete an element by right clicking on it and selecting *delete element* from the pop-up menu, or you can select the element by left clicking on it and pressing the delete key.

If you want to delete many elements, you can select multiple elements by holding down the shift key while left clicking on them, or using a "rubber band" box to select multiple elements at once.



Once the multiple elements are selected, press the *delete* key to delete them.

Rather than deleting the elements, you can use the right click menu to copy the elements and then paste them into either the same model, or a different model in a different Modeler instance.

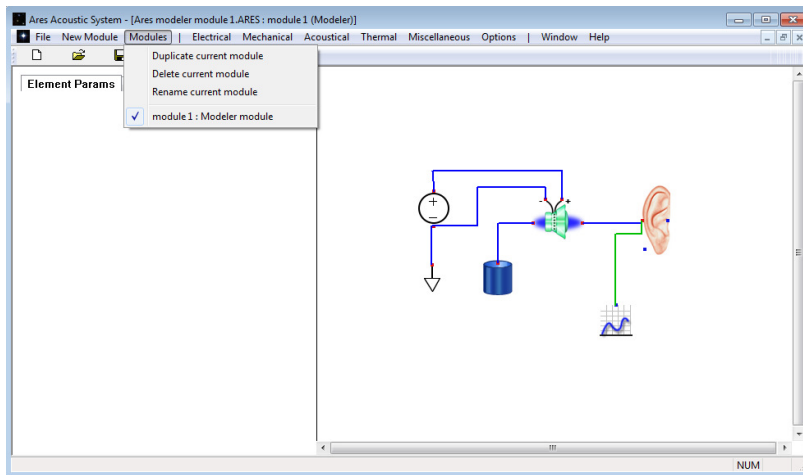
Duplicating Models

An extremely useful feature of Ares is the ability to duplicate modules. This is extremely useful when performing a design and you want to try alternative parameter values or a different element arrangement.

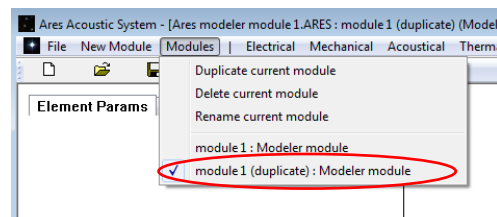
Suppose we want to try and improve our earpiece design by adding a leak between the speaker and the ear, but we don't want to modify the original model design because we're already fairly

happy with it. We just want to see if we can improve upon it, but we want to keep the original design intact.

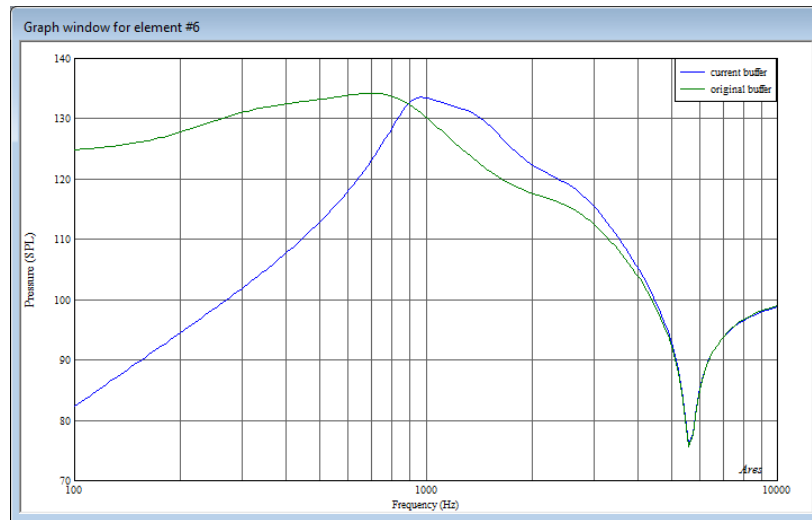
By selecting *Duplicate current module* from the *Modules* menu, an exact duplicate of the current module is created.



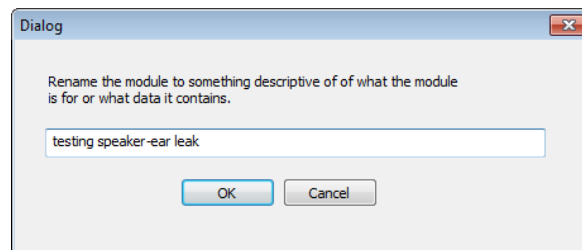
The list of module instances are displayed at the bottom of the *Modules* menu. The new module instance will have the name of the original module instance with "(duplicate)" appended. Since the original model just had the default name of "module 1", the new module's name is "module 1 (duplicate)".



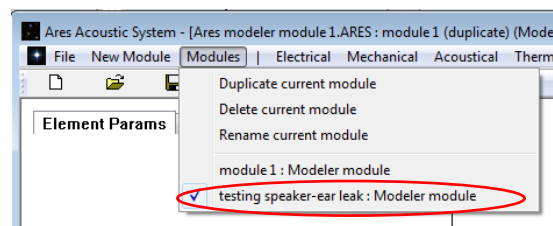
Now select *radiating rectangular port* from the *Acoustical* menu and place the new element as shown below. Make sure you connect the red connector on the new element to the front of the speaker. This simulates a leak to the outside air. Press F5 to solve the model. The result is shown below. Note that the main result is that the low frequencies are lost, as they easily escape through the rectangular port due to its low impedance at low frequencies.



Now that we've modified the model, we want to give this new model instance a name that will be meaningful later on. By selecting *Rename current module* from the *Modules* menu, the following dialog box will appear that will allow you to change the module instance's name. Change the name to "testing speaker-ear leak" and press OK.



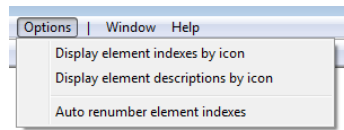
Now when the *Modules* menu is pulled down, the name of the current module will be changed to a meaningful name.



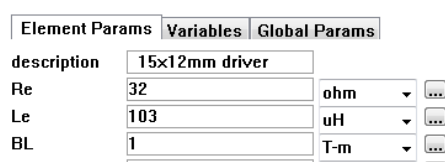
While the above model and its modification was trivial, as the user gains experience with Ares, he/she will be generating much more complicated models and designs. The user will find this ability to duplicate models to create a design "branch" of their main design extremely useful. Creating a duplicate model is useful even to test some of the model's parameter changes. Make sure you keep this method of investigating different designs in mind as you use Ares.

Options Menu

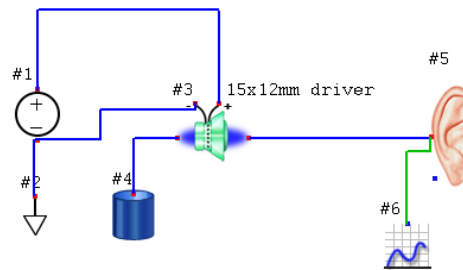
The Options menu controls the text that's displayed above the elements. Each element is given an index number, starting at 1. These indexes make it easier to denote an element when describing the model. By selecting *Display element indexes by icon*, the element number will be displayed.



Also, note that the first parameter for every element is a description string. Below are the first few parameters for the speaker element.



The description strings can add descriptive information about the elements or how they affect the model. Below the model is shown with both the indexes and description text displayed.



Node types

There are two types of nodes: driven and sensing. Most of the nodes are driven nodes, which appear as red squares on the icons. An example of a driven node is the single node of a volume element. This node is driven by the rest of the circuit, and will load down the circuit. Since the node loads the circuit, changing the volume element's parameters will change the model's solution at all of the other nodes.

A sensing node appears as a blue square on the icons. These nodes are not to be driven, but are outputs that produce frequency dependent values computed from the element's parameters, and/or from the element's other nodes. An example of a sensing node is the far right node on the radiation elements. The radiating circular port is shown below. The value of the sensing node on the right is determined by the red node on the left which is driving the radiating port. The radiation part of the element predicts the pressure at a distance from the port, assuming that the sound radiated from a baffled circular port. The blue sensing node on the right represents this

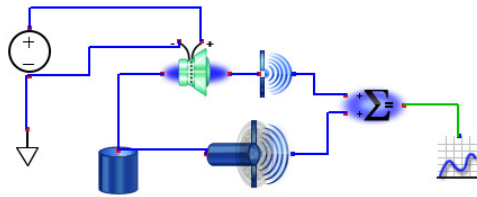
radiated pressure. If the blue node is connected to other acoustical elements, those element will not change (or load) the value at the sensing node.



The radiated pressure is computed from, and is completely defined by the volume velocity out of right end of the circular port. If another element is connected to the blue sensing node, that element will **not** load down the rest of the model. This is because a sensing node's pressure (or voltage, etc.) is only a function of the solution at the driving nodes. In general, you either connect a graph element or a summation element to the sensing nodes as it generally doesn't make sense to connect additional acoustical elements to sensing nodes.

Another important element that uses sensing nodes is the ear simulator. Its sensing nodes display the DRP and free field corrected frequency responses. This is discussed further in the elements section.

For instance, suppose you want to model a bass reflex speaker design. This would involve a speaker with a rear volume, a radiating element on the front of the speaker, and a radiating port attached to the rear volume. To see the total pressure from the front and bass reflex port, you would add the two pressures from radiation sensing nodes together. This is shown below.

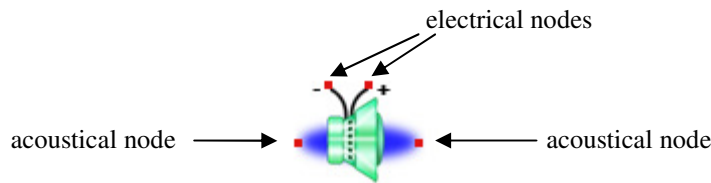


Note that it generally wouldn't make sense to add a volume element to the radiating element's sensing node. But if you were to do so, you'd find that the volume element would not change the solution at the sensing node what-so-ever.

Also, for a node, the "kinetic values" such as current, velocity, volume velocity and thermal flow are always defined to be INTO the node. This is particularly important to understand when you're plotting impedance data. If you want to see the impedance that the rear of the speaker is driving, you can put the graph connector on the rear speaker node and then select *plot using negative kinetic value* from the graphs *Advanced Options* dialog box. This will essentially change the impedance quantity from looking into the node to looking out of the node.

Node Domain Types

For the most part, the elements are one of four domain types: electrical, mechanical, acoustical or thermal, and you can only connect similar element types to the same type. That is, you cannot connect an electrical element to an acoustical element. However, some elements are actually hybrids. One common example of this is the speaker element. It has both electrical and acoustical nodes.



In this case, you can only connect the acoustical nodes to other acoustical nodes, and likewise with the electrical nodes. Ares will not let you connect an electrical node to an acoustical node.

The icons have been designed so the node's domain type is shown by the color of the field that the node appears in. In general, if the connection node is in:

- a white field it's an electrical node
 - a green field is a mechanical node
 - a blue field is an acoustical node
- and
- a red field is a thermal node.

This is shown below for the electrical voltage, mechanical force, acoustical circular port, and thermal mass elements. However, these elements don't mix node types.



The elements below are hybrid elements that mix electrical, mechanical, and acoustical node types. The first one is a BL coupler that converts an electrical signal to a mechanical force. This simulates the behavior of a moving coil speaker's magnetic motor structure. Note that the two electrical nodes are in a white field, and the mechanical nodes are in a green field. The last element (on the right) is an omni microphone. It has an acoustical node that's in a blue field, and two electrical nodes (above and below the icon) that are in a white field.



Again, you can only connect electrical nodes to other electrical nodes, mechanical nodes to other mechanical nodes, etc.

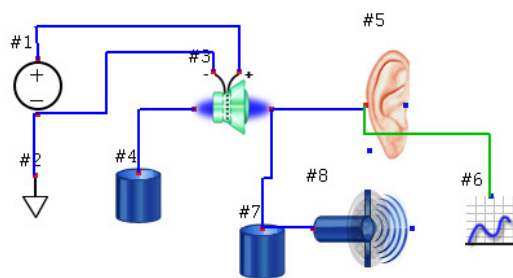
When a circuit simulator such as SPICE is used to simulate mechanical/acoustical lumped parameters, an inductor is used not only for the electrical element, but also for mechanical and acoustical masses. The expert user has to keep track of what node is being used for each domain (electrical, mechanical, etc.) and apply the correct conversion when graphing results. Ares keeps track of this automatically.

One final note on nodes: If you don't connect a node to anything, a zero velocity or current will be assumed. An exception to this is for mechanical elements which will assume a zero force. This is because the mechanical domain is solved using an admittance model, whereas the electrical and acoustical domains are solved using impedance models.

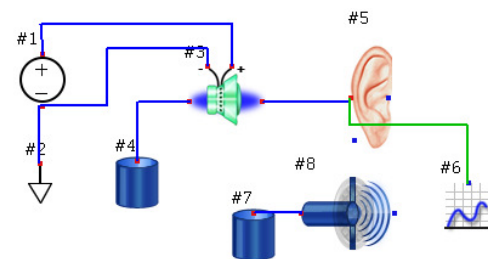
Disabling model sections

Occasionally you'll want to temporarily remove some sections of the model. Perhaps you're experimenting with adding additional elements, or you want to identify just how some of the elements are affecting the model response.

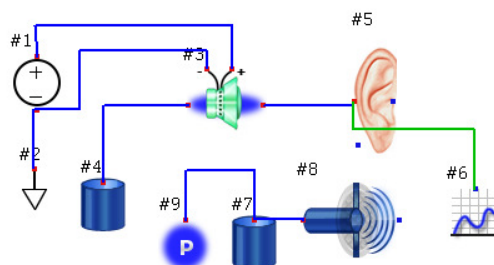
To disable such sections, you need to remove the connectors from the elements in question from the main part of the model. Consider the model below. Suppose you want to see the effect of removing the volume element #7 and the radiating circular port #8 from the model.



You can do that by simply removing the connector from the volume element #7 to the speaker element as shown below. So now the pressure and velocities for elements #7 and #8 will all be zero.

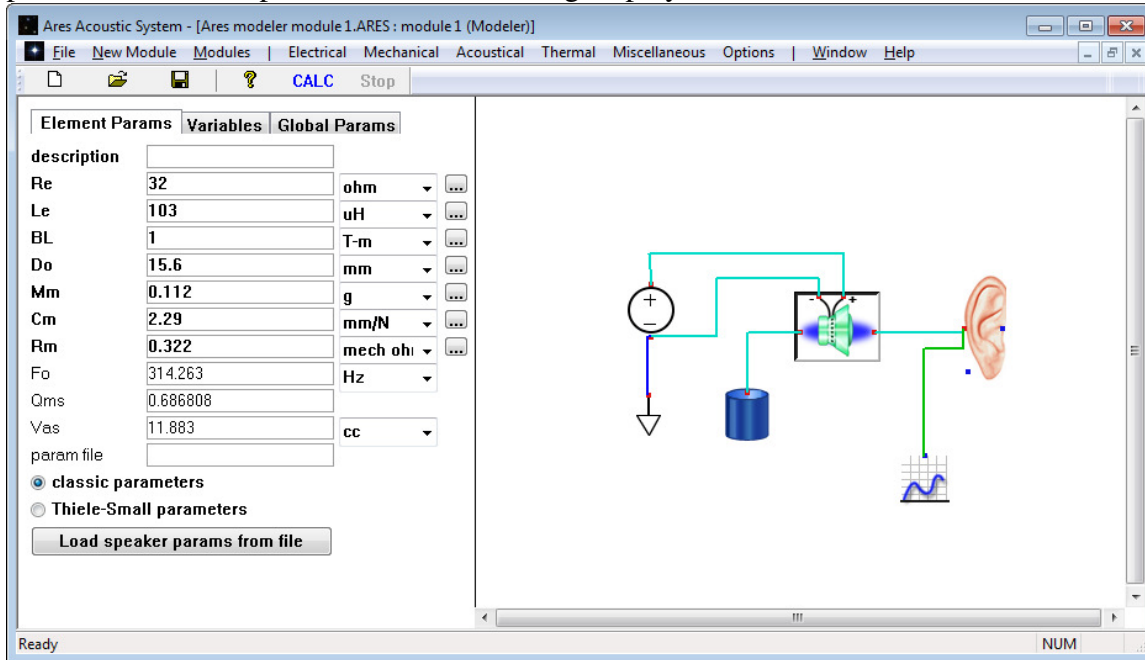


However, simply disconnecting elements and leaving them undriven will sometimes cause the solver to generate a singular solution error for the system. When this happens, you can avoid the singular solution by simply driving the disconnected elements with an appropriate element. Below shows a pressure source element #9 being used to drive the disconnected elements, which will avoid a singular solution error.




Modifying Element Parameters

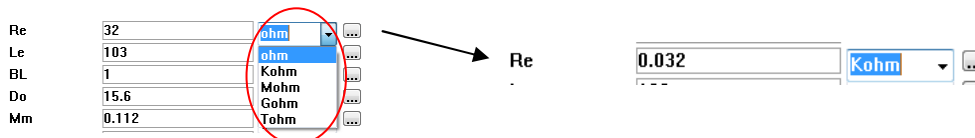
When you click on an element to select it, its parameters will be displayed in the parameter area on the left side of the Ares window (you'll need the *Element Params* tab selected). The parameters for the speaker element are being displayed for the model below.



The unique aspects of the parameter values are:

- the units can be changed by using the drop down list box to the right of each parameter
- an equation can be entered for their value
- variable names can be used in the value equations
- the parameter value can be "iterated" upon by pressing the  button to the right of each parameter

An example of changing the units is shown below, where the speaker parameter Re has had its units changed from ohm to Kohm. Note that Ares will automatically scale the numeric value to the appropriate units, so 32 ohms becomes 0.032 Kohm.



An example of using an equation is shown below, where the speaker diameter Do is being computed from a radiating surface area of 200 mm². Since the area $A = \pi (Do/2)^2$, $Do = 2 * \sqrt{A / \pi}$. This also shows how a variable can be used in the parameter list. In this case, the built in variable "pi" is being used to represent 3.14159265.

Re	0.032	Kohm	▼	...
Le	103	uH	▼	...
BL	1	T-m	▼	...
Do	$2*\sqrt{200/\pi}$	mm	▼	...
Mm	0.112	g	▼	...

It's important to understand that when the equation is evaluated, the numeric value will assume the units specified to the right of the parameters value. In this case, $2*\sqrt{200/\pi}$ evaluated to 15.96 which will have units of mm. Had the units for Do be set to m, the Do would have a diameter of 15.96 meters!

In general, the syntax used for evaluating variables follows Matlab. The following operators and functions are presently implemented. (If more functions are needed, you can request they be added to the next version of Ares.)

Operators

:	range operator
,	matrix element separator
+	plus
*	multiply
.*	element by element multiply
/	divide
./	element by element multiply
\\	solve linear system
^	exponent
.^	element by element exponent
~	not
==	equals
~=	not equal
&	and
<	less than
<=	less than or equal to
>	greater than
>=	greater than or equal to
.	transpose
'	Hermitian
end	last element of an array

Functions

find	returns an array of non-zero elements
sin	sine (radian argument)
cos	cosine (radian argument)
tan	tangent (radian argument)
asin	arc sine (returns radians)
acos	arc cosine (returns radians)
atan	arc tangent (returns radians)
atan2	arc tangent of y,x (returns radians)
db	decibel (same as $20*\log_{10}(\text{abs}(x))$)

abs	absolute value
angle	angle (returns radians)
log	log base e
log10	log base 10
exp	exponent
mod	modulus
conj	complex conjugate
sqrt	square root
%	comment


Built in variables:

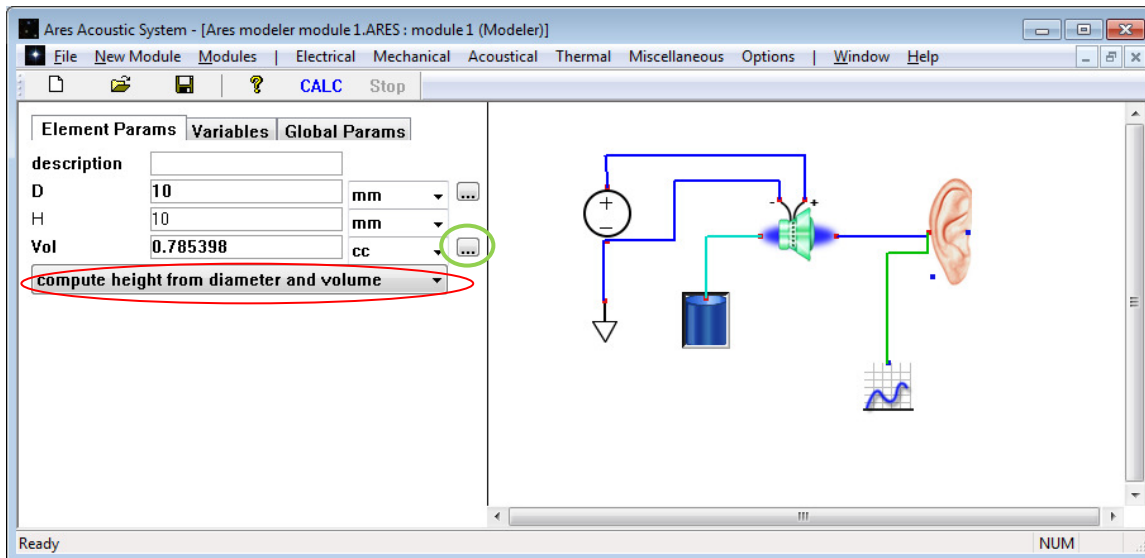
i	$\sqrt{-1}$
j	$\sqrt{-1}$
pi	3.14159265
f	frequency (Hz)
jw	complex radian frequency = $j*2*\pi*f$


Iterating a Parameter

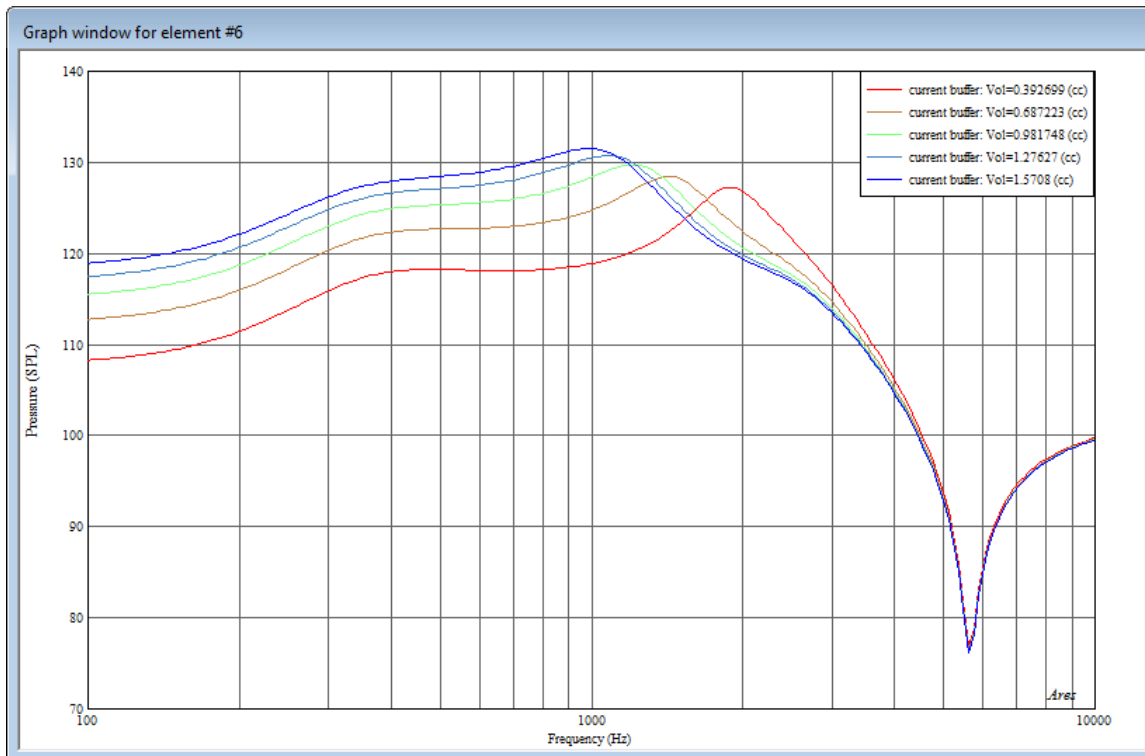
Iterating a parameter means to run the model for a range of parameter values. For instance, if you iterate upon a port length, the model will be automatically solved multiple times for a variety of port lengths. Each node will then contain "multiple solutions" that will be plotted when a graph element is attached to a node. Iteration results can be saved in a graph buffer just as a single model solution.

By default, the range over which the iteration is performed is from 50% of the current value to 200% of the current value, or from one half to times two. Also, by default, 5 iterations will be performed. However these ranges can be changed in the *Global Params* tab section.

For the model below, we'll iterate upon the rear volume. To iterate upon the volume directly, and not the height or diameter, we'll change the calculation option to *compute height from diameter and volume* as shown with the red oval. Next, press the iteration button  to the right of the *Vol* parameter as shown with the green oval.



The result of pressing the iteration button  for the Vol parameter results in the solution shown in the graph below. Note that the actual volume value used for each curve is shown in the legend. The result is that the low frequency output drops as the volume is decreased.



Parameter iteration is a good way to optimize a design. For instance, the above graph shows that volumes much larger than 1 cc start to have diminishing returns on increasing the low frequency output.

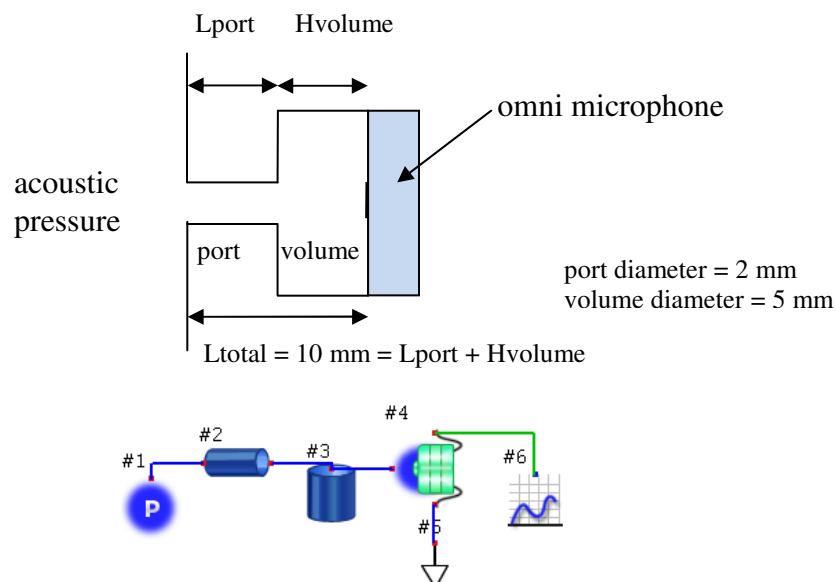
Parameter iteration is also a good means of determining what elements are controlling the response, especially around resonances or dropouts. For instance, the above model has a significant dropout at around 5.5kHz, but the response at that frequency didn't change at all when the rear volume was iterated. This indicates that the rear volume is not controlling or having any significant effect on this drop out. The designer would know to look elsewhere to improve the response near this frequency.

To adjust the iteration range, select the *Global Params* tab. The iteration parameters are located at the bottom of the controls as shown below.

Element Params	Variables	Global Params
Frequency range for analysis		
Fmin	100.0	Hz
Fmax	10000.0	Hz
num points	200	
<input checked="" type="checkbox"/> logarithmic frequency distribution		
Iteration parameter range		
min iteration	50.0	%
max iteration	200.0	%
num	5.0	
Fluid properties		
<input checked="" type="checkbox"/> compute fluid parameters for dry air		
Temperature	73.0	F
rho (density)	1.1847	kg/m ³
B (bulk)	1.4201e+005	Pa
gamma (ratio)	1.4018	
mu (shear)	1.8372e-005	N s/m ²
k (thermal)	0.0261	W/m K
cp (heat)	1006.0	J/kg K
<input type="checkbox"/> use porous material model		
pore diameter	0.2	mm
porosity	0.99	
tortuosity	1.0	

Variables

Variables allow you to specify critical parameter values in one central location and to define mathematical relationships between them. This allows for very useful parametric studies. For instance, suppose you have a microphone port design where the length of the port and the height of the volume in front of the speaker must add up to 10mm. This is sketched below.



The design constraint is that the length of the port and the height of the volume must add to 10 mm. To model this, we're going to create three variables. First, select the *Variables* tab and click on *New variable* three times. Change the name of the first variable to Ltotal and give it a value of 10, change the name of the second one to Lport and we'll start with a value of 1 for it, and then the third will be named Hvol and we'll set it to Ltotal-Lport as shown below.

Element Params Variables Global Params				
New variable				
Ltotal	10	...	ins	del
Lport	1	...	ins	del
Hvol	Ltotal-Lport	...	ins	del

The port parameters (element # 2 in the model) are assigned as shown below. Note that for the length, we've used the variable Lport instead of a numeric value.

Element Params Variables Global Params				
description				
L	Lport	mm	...	
Do	2	mm	...	
R	0	MKS Ray	...	
Num	1		...	
Vol	0.00314159	cc		
S	3.14159	mm ²		
Ma	380.015	kg/m ⁴		
compute Ma from length and diameter				

The volume parameters (element # 3) are assigned as shown below. Note again that for the height, we've used the variable Hvol instead of a numeric value.

Element Params Variables Global Params				
description				
D	5	mm	...	
H	Hvol	mm	...	
Vol	0.176715	cc		
compute volume from height and diameter				

The default value for the pressure source (element #1) is 94 dB, which corresponds to 1 Pa. However, we're going to change it to 94+42 dB as shown below. The reason for this is that the default sensitivity of the microphone is -42 dBV/Pa. A pressure value of 94+42 will result in 0 dBV out of the microphone assuming the port and volume structure do not affect the microphone response.

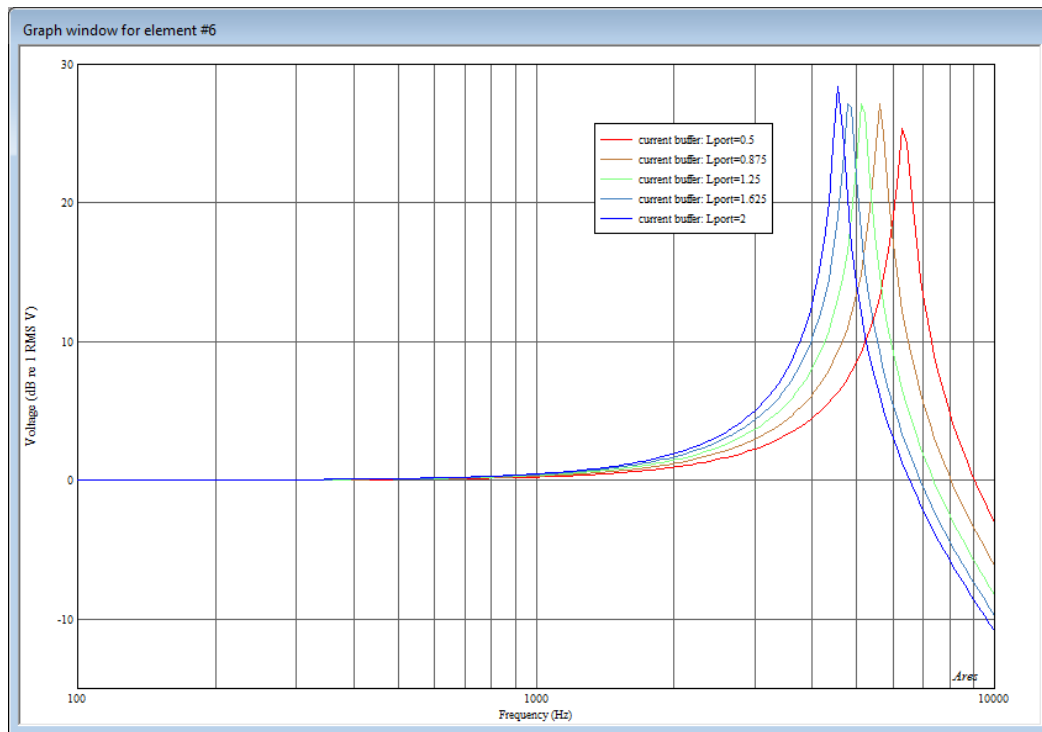
Element Params Variables Global Params				
description				
P	94+42	SPL	...	

Finally, we've just used the default parameter values for the omni microphone (element #4).

By pressing the iteration button for the Lport variable as shown below, we can compute a parametric set of frequency responses of the system for a variety of Lport values, while still maintaining the Ltotal=10mm design constraint. This is shown in the graph below.

Element Params Variables Global Params				
New variable				
Ltotal	10	...	ins	del
Lport	1	...	ins	del
Hvol	Ltotal-Lport	...	ins	del

Note that since the low frequency sensitivity of the response is 0 dBV, the levels at the high frequencies can be read as an effective increase in sensitivity due to the port/volume resonance. (A 20-30 dB peak in the response is typically extremely undesirable, so if this microphone is meant to operate over 3kHz, this design would not work.)



Global Parameters

The *Global Params* tab displays a set of parameters that are common to the entire model and the fluid properties which are common to the circuit layer. We've already seen the iteration parameters, but these parameters also include the frequency range as well as the air properties used for the elements.

Element Params	Variables	Global Params
Frequency range for analysis		
Fmin	100.0	Hz
Fmax	10000.0	Hz
num points	200	
<input checked="" type="checkbox"/> logarithmic frequency distribution		
Iteration parameter range		
min iteration	50.0	%
max iteration	200.0	%
num	5.0	
Fluid properties		
<input checked="" type="checkbox"/> compute fluid parameters for dry air		
Temperature	73.0	F
rho (density)	1.1847	kg/m ³
B (bulk)	1.4201e+005	Pa
gamma (ratio)	1.4018	
mu (shear)	1.8372e-005	N s/m ²
k (thermal)	0.0261	W/m K
cp (heat)	1006.0	J/kg K
<input type="checkbox"/> use porous material model		
pore diameter	0.2	mm
porosity	0.99	
tortuosity	1.0	

F_{min} , F_{max} , and *num points* determine the frequency range and the number of frequencies to be solved for. The *logarithmic frequency distribution* check box determines how those points are to be distributed from F_{min} to F_{max} . Either logarithmic (if checked) or linear (not checked).

As previously discussed, the *Iteration parameter range* determines how the parameters are iterated upon. They operate from a fraction (in percent) of the current value to another fraction. By default, the range goes from 50% (or one half the current parameter value) to 200% (twice the current parameter value) with 5 steps. The author has found this range to be the most useful for most cases. An exception is for directional microphone modeling, which will be discussed in the directional microphone example.

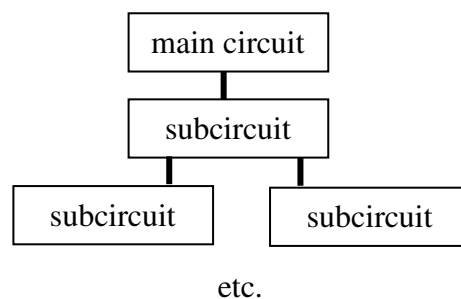
The fluid properties determine the properties of the air in the elements. By default, the parameters will be computed for dry air at 73 degrees F. However, by unchecking *computer fluid parameters for dry air*, the fluid properties (such as density, modulus) viscosity, etc.) can be entered manually. For instance, for underwater acoustics, the parameters for water can be entered. However, most users will not be changing the fluid properties.

There also is a "porous material" option which will simulate filling the ports and volumes with a porous material such as polyester, fiberglass, or open cell foams. Porous materials are typically used to increase the effective volume behind a speaker by causing the air to behave isothermally. There are videos on the MAE website that details this behavior. <http://maellc.com/videos.asp?vid=AM5#AM5>

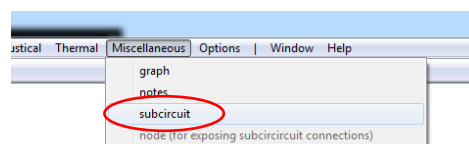
The fluid parameters are applied to all elements in the circuit level and by default will be applied to all subcircuit levels. To apply different fluid parameters to elements, those elements must be placed in a subcircuit.

SubCircuits

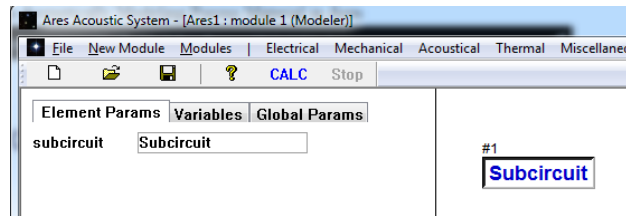
When a model becomes too complicated, or if you want to apply different fluid parameters to the acoustic elements, a sub-circuit can be used. A subcircuit adds a new layer to a model. Any number of subcircuits can be used. The subcircuit model is shown below.



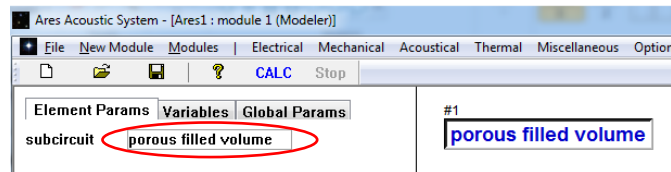
To create a sub-circuit, select the *subcircuit* command from the *Miscellaneous* menu and place the element in to the model.



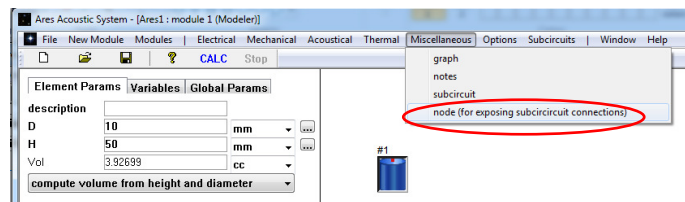
By default, the subcircuit will be given the name "Subcircuit" as shown below.



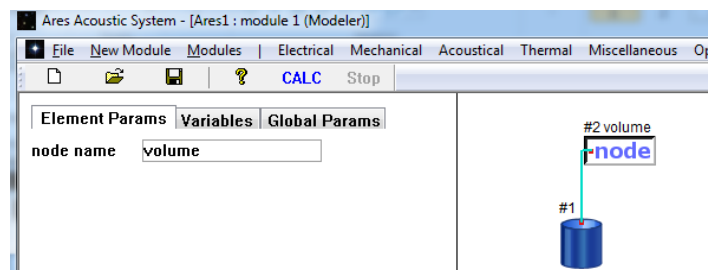
But it can be renamed by editing the field when the subcircuit element is selected as shown below.



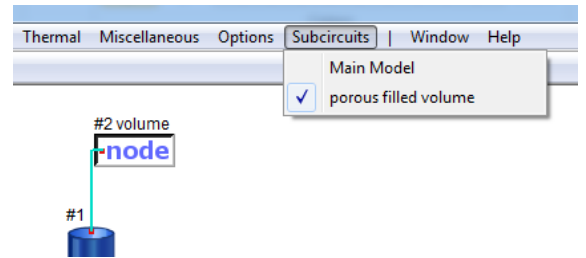
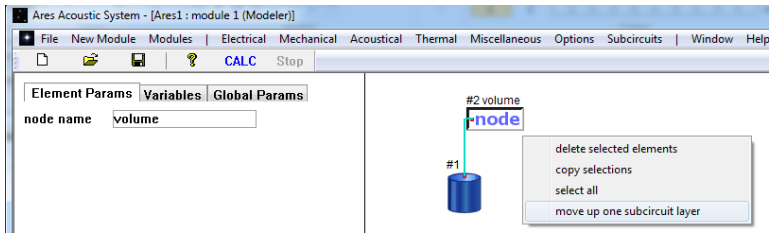
To edit the subcircuit, double click on the icon and add elements to the model. The model below is a simple volume. To connect nodes in the subcircuit to the upper circuit level, a "node" element needs to be created by using the *node (for exposing subcircuit connections)* command from the *Miscellaneous* menu as shown below.



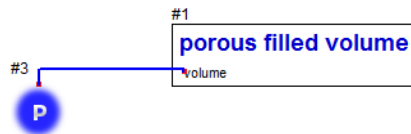
The node elements can then be given a name and connected to the subcircuits. In the model below, the node name is "volume" and it has been connected to the volume node.



To return to the upper circuit level, you can either use the right click menu and select *move up one subcircuit layer* or use the *Subcircuits* menu and select the circuit level to move to.

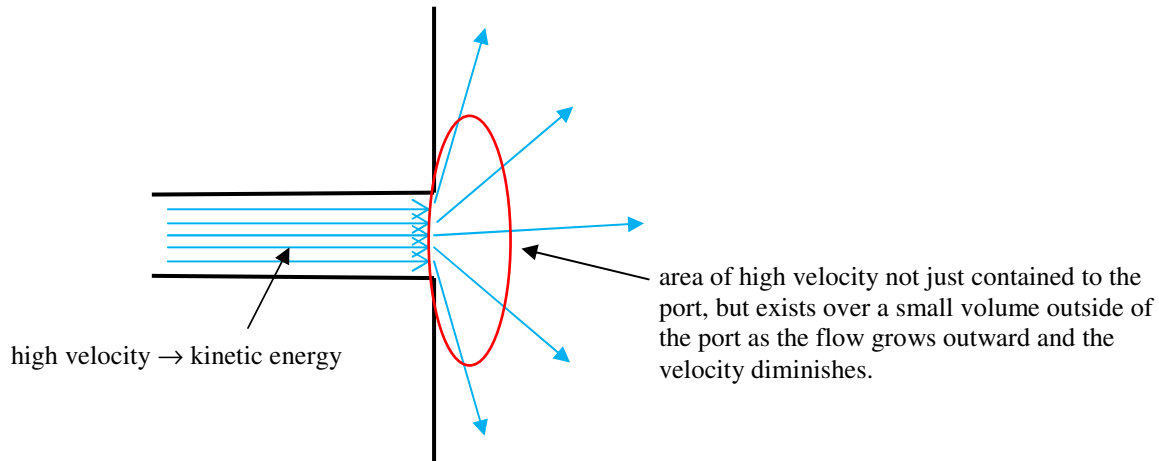


Once you return to the upper circuit layer, the nodes that were created in the subcircuit will appear in the Subcircuit icon as a connection node with the name given in the subcircuit. This node can be connected to the circuit and will appropriately load the circuit.



End Corrections

Acoustic ports store kinetic energy in the air velocity, or momentum. When the acoustic wave exits a port, the wavefront opens up gradually. One of the effects of this is that there is a significant amount of energy stored in the air velocity in the vicinity of the port.



The kinetic energy present just outside of the port is frequently referred to as an "attached mass". For a circular port, this mass is directly proportional to its radius and is commonly added to the length of the port. For a port of length L and radius a , the "effective length" of the port is the physical length of the port plus an end correction. The end correction depends on how the sound interacts with the environment when it exits the port. For the two common radiating conditions of baffled and unflanged, the effective lengths are:

$$L_{\text{eff}} = L + 0.85 * a \quad \text{for a baffled port}$$

$$L_{\text{eff}} = L + 0.6 * a \quad \text{for an unflanged port}$$

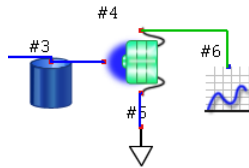
where a baffled port is a port radiating from an infinite plane or wall, and an unflanged port is the end of a tube without any wall structure around it (like the end of an organ pipe or a flute).

For a long port with a small diameter, the end correction is negligible. However, for a very short port, or for a port that's just a hole in a thin wall, the end correction can dominate its acoustical behavior.

So to accurately calculate the response of an acoustical model, the end corrections associated with the acoustical element must be accounted for. Ares uses a proprietary algorithm for computing a good estimate for the end correction for the acoustical elements. This eliminates the need for the user to manually enter an end correction for each element, and it eliminates errors associated with not using them at all, which novice users are prone to do.

Graph Element

The graph element is the sole mechanism for getting access to the model results. Each graph element creates a separate window on the desktop that displays the actual graph. You are not limited to one graph, you can create as many graphs as you like. The graph element has one sensing connector node. When the graph is connected to a model node, the data from that node will be copied into the graph element's *Connector node* buffer. For the model below, the electrical data for the top microphone node will be copied into the graph element for plotting.



To effectively use the Graph element, it's important to understand the buffer scheme. The main graph parameters are shown below. The check box under the *Connector node* indicates that the data in the *Connector node* buffer is to be plotted. The graph element contains multiple buffers that can each store model node solution data. There are 26 buffers, labeled A-Z. They are displayed in groups: A-I, J-R and S-Z. When one of the graph buffer buttons such as **A**, **B**, etc. are pressed, the data from the *Connector node* buffer will be copied to that buffer, A-Z. Data in that buffer will be plotted in the graph window if the checkbox to the left of the buffer button is selected. Descriptions for each buffer can be entered which will appear in the graph legend.

press a button to copy data into the buffer

select a check box to plot the data in that buffer

enter descriptions for the plotting buffers, the descriptions will appear in the graph legend for that buffer data

Element ParamsVariablesGlobal Params

description

Connector node

☒

Plotting buffers

☐ A

☐ B

☐ C

☐ D

☐ E

☐ F

☐ G

☐ H

A-I

J-R

S-Z

quantity

voltage

type

log x

☐ vs iterator @Hz

0

☐ normalize to y=

0

at x=

0

x axis range (min

y axis range (min

Update graph

Hide graph

Set buffer line

Advanced Options

Each node contains two different types of data that depends on the node type as indicated by the table below.

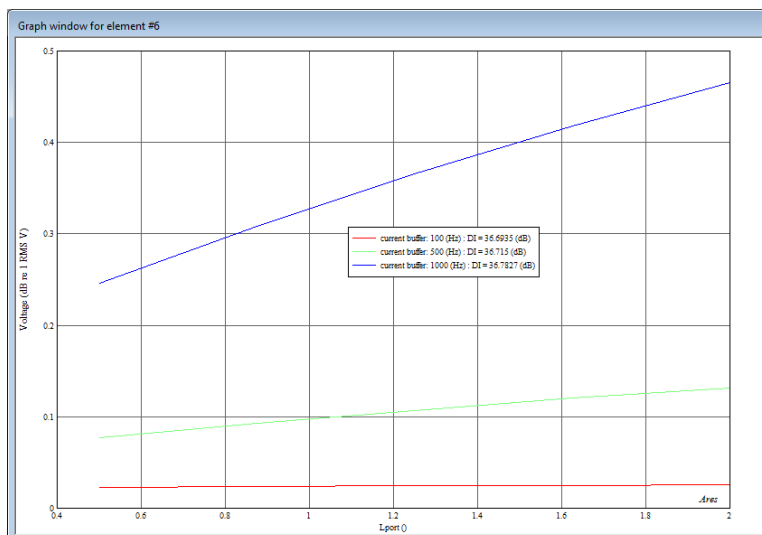
Node Type	Quantities
electrical	voltage and current
mechanical	force and velocity
acoustical	pressure and volume velocity
thermal	temperature and heat flow

The *quantity* list box indicates what quantity is plotted. The graph element will perform useful conversions for appropriate data. For instance, you can not only plot the volume velocity for an acoustic node, but you can also plot the particle velocity and displacement. Particle velocity is the volume velocity divided by the cross section area associated with the node, and displacement is the frequency domain integration ($1/j\omega$) of the particle velocity. When data from multiple node types are plotted simultaneously, such as electrical and acoustical, the units from the first buffer plotted will be used.

The *type* listbox controls whether the graph is plotted with log or linear scales.

The *vs iterator @Hz* checkbox allows you to plot data vs an iterator value for specific frequencies. This is only useful for buffers that contain iteration data. An example of this is show below where the iteration data is plotted for 100, 500 and 1000 Hz. This is for the previous microphone model where the x axis is the Lport value. Plotting vs the iterator value is especially useful for performing microphone directionality designs which will be demonstrated in the microphone example.

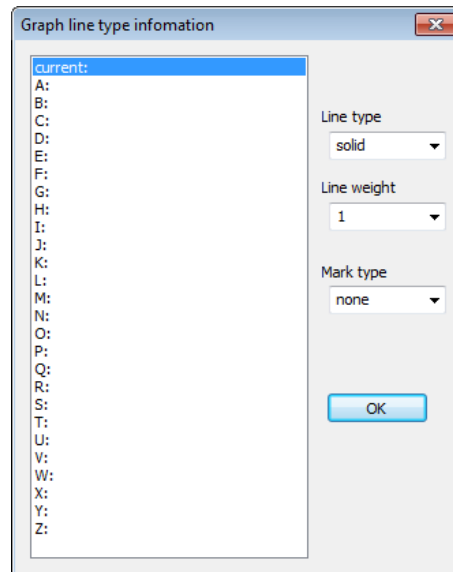
☒ vs iterator @Hz 100 500 1000



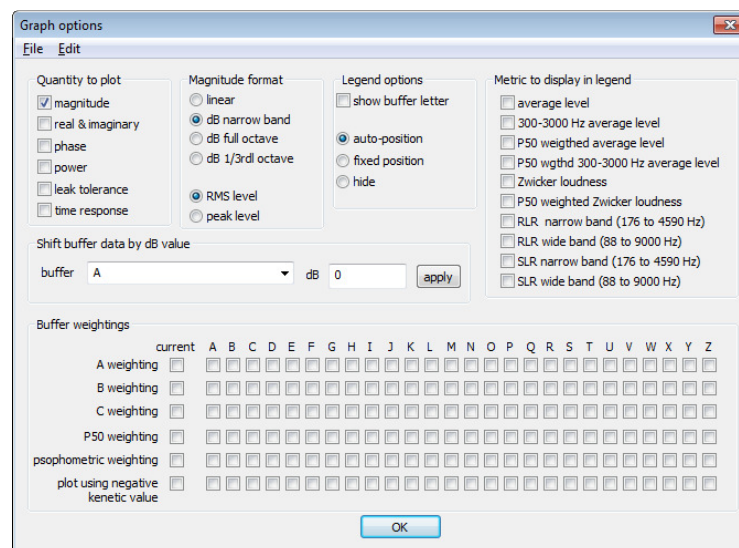
The *Update graph* button forces the graph to be redrawn with new settings.

The *Hide graph* check box causes the graph window to be closed, removing clutter from the desktop.

The *Set buffer line* button causes a dialog box to appear that allows you to set the line type, weight and mark for each buffer curve. Color is generated automatically.



The *Advanced Options* button causes the following dialog box to appear. This has many useful features to help present the node data in a useful format. Any changes to the entries in this dialog box are immediately applied to the graph. You do not need to close the dialog box to see the effects.



Quantity to plot

By default, only the magnitude of the node data is plotted, however, the node data is complex valued. Selecting more plot types will cause them to appear in the graph window. Leak tolerance plots the difference between Type 3.2 low leak and Type 1 and 3.2 high leak ear simulation results. This requires data that has iterated using these ear simulators. *Time*

response performs an FFT on the frequency data to produce a time domain data plot to show an impulse response.

Magnitude format Determines how the magnitude data will be plotted, either as a linear quantity, dB, or in octave bands. Also, peak or RMS levels can be plotted. By default, Ares assumes RMS drive units for all voltages and pressure sources.

Legend options The buffer letter can appear in the graph legend if desirable. Also, by default the legend can be dragged around. You can fix its location or opt for it not to be drawn at all.

Metric to display in legend

There are many metrics that can be displayed in the graph legend for each buffer. Ares will do its best to make sense of the node data and generate the appropriate metric requested.

average level	a power average of the frequency data
300-3000 Hz average level	power average over the 300 to 3000 Hz data
P50 weighted average level	power average after a P50 speech weighing is applied
P50 wghd 300-3000Hz average level	power average over 300 to 3000Hz data after a P50 speech weighing is applied
Zwicker loudness	Zwicker loudness computed from frequency response
P50 weighted Zwicker loudness	Zwicker loudness computed from P50 weighted frequency response data
RLR narrow band (176 to 5490 Hz)	ITU RLR (Receive Loudness Rating) computed from spectrum data, valid on spectrum with units of Pa/V
RLR wide band (88 to 9000 Hz)	Wideband ITU RLR (Receive Loudness Rating) computed from spectrum data, valid on spectrum with units of Pa/V
SLR narrow band (176 to 5490 Hz)	ITU SLR (Send Loudness Rating) computed from spectrum data, valid on spectrum with units of V/Pa
SLR wide band (88 to 9000 Hz)	Wideband ITU SLR (Send Loudness Rating) computed from spectrum data, valid on spectrum with units of V/Pa

Shift buffer data by dB value

This feature will multiply the node data by the associated dB scaling factor and permanently change the data in the graph buffer. This is especially useful when you've imported measured data that needs to be properly scaled to better match the model data. Note that since the impedance is a ratio of the pressure and velocity (or voltage and current, etc.), this dB scaling does not affect impedance data.

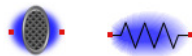
Buffer weighting Multiple weighting types can be applied to the data presented for each buffer. This large matrix of check boxes allows you to determine which weighting is to be applied to which buffer.

The *plot using negative kinetic value* is unique. It changes the sign of the current, or velocity, or thermal flow values. One of the results of this is that impedances will change sign. Since all "kinetic values" are defined to be into the node, this is useful when you want to plot the impedance looking out of a node.

Acoustic Material Database

The impedance of an acoustical device, such as a porous material, membrane, or even a complex shaped port, is not always simply modeled from a few parameters. In these cases, it is sometimes easier to simply measure the impedance of such devices. Ares' Flow Impedance Module provides such impedance measurements. For instance, suppose you have an acoustic material that covers a speaker to keep out water and dust. The effect of that material will be described in terms of its acoustical flow impedance it presents to the sound that comes from that speaker.

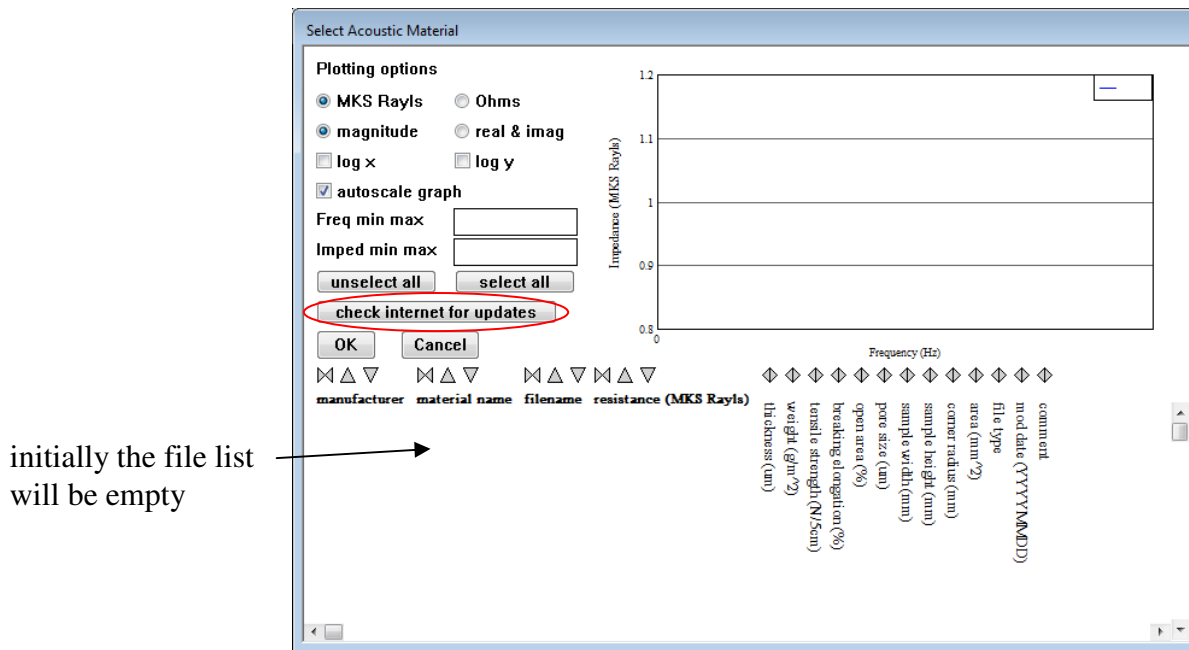
Ares has two elements that can load in acoustical impedance data and use it directly in their simulations. These are the *acoustic material* and *resistor* elements, whose icons are shown below.



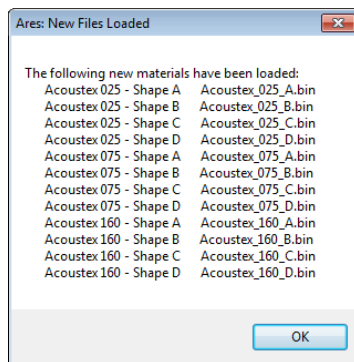
The parameters for the *acoustic material* element are shown below. The *Load impedance data file* button allows you to load impedance data into the modeler directly from a file. Once the data has been loaded, the *use file impedance data* check box will be enabled. Once checked, the *R* parameter, representing the resistance of the material in MKS Rayls, will be ignored and the frequency dependent impedance data from the file will be used in subsequent simulations. This is a very useful means to use custom measured or computed impedance data directly in an Ares model. (Note: the *acoustic material* element uses MKS Rayls impedance data, while the *resistor* uses Acoustic Ohms impedance data, where Acoustic Ohms is pressure over volume velocity, or Pa-s/m³.)

Element Params	Variables	Global Params
description <input type="text"/>		
R	<input type="text" value="30.0"/>	MKS Ray <input type="button" value="..."/>
Do	<input type="text" value="10.0"/>	mm <input type="button" value="..."/>
Num	<input type="text" value="1.0"/>	<input type="button" value="..."/>
<input type="button" value="Select materials from database"/>		
<input type="checkbox"/> use database impedance data		
No materials selected <input type="button" value="..."/>		
<input type="checkbox"/> use file impedance data		
<input type="button" value="Load impedance data file"/>		

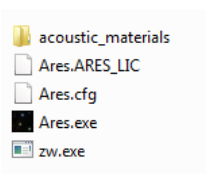
MAE provides another way to get impedance data files using an internet database established on the MAE web site. You download files from the database which get stored to a local directory. To gain access to the data files in the online database, press the *Select materials from database* button. The screen below will then appear. The screen shows the local files which initially will be empty.



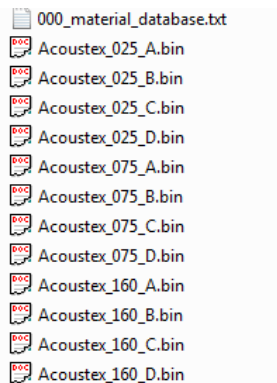
To check the internet for new files and populate your local directory, press the *check internet for updates* button. If new files are found, they will be listed in a dialog box that will appear after the internet database is checked.



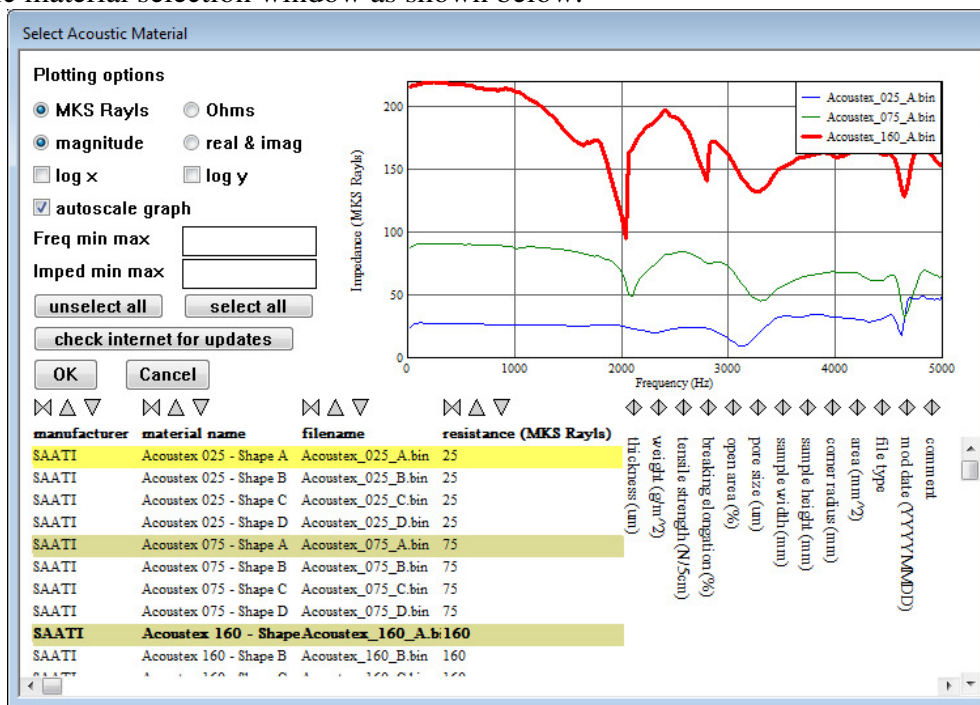
Any new files will be downloaded to a subdirectory called *acoustic_materials* of the directory that Ares.exe is being run from. The contents of the Ares.exe directory is shown below with the *acoustic_materials*.



The contents of the newly created *acoustic_materials* directory is shown below. The new impedance data files appear as *.bin custom binary format files with the 000_material_database.txt file giving the master list of these files for Ares. It is recommended that you not make any changes to this directory. If it becomes corrupted, you can delete the *acoustic_materials* directory and recheck the internet for updates. Ares will recreate the directory for you with all of the impedance data files from the internet database.



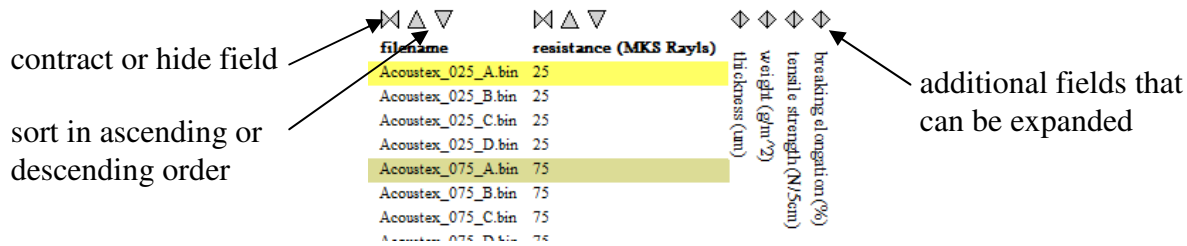
Once the local *acoustic_materials* directory is populated with data files, they will appear in the acoustic material selection window as shown below.



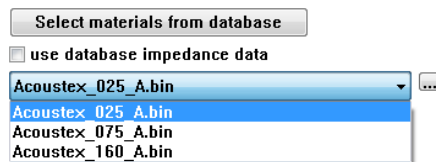
The interface tries to pack a lot of information into a small space. By clicking on an impedance file (shown in yellow), you can plot its impedance in the upper right corner of the window. By double clicking on a file, you can select it for importing into the element for simulation. Multiple files can be selected and brought into the element. You'll be able to either select one of these files to use in the element, or iterate upon all of the selected files just as you'd iterate on a numeric parameter. (The file list can also be navigated by using the up/down arrows, and the spacebar for selecting.)

The upper left corner of the window allows the impedance data to be plotted in different units, in magnitude or real and imaginary, and log or linear scales. The graph can also be either auto scaled, or manually have the x-y ranges entered.

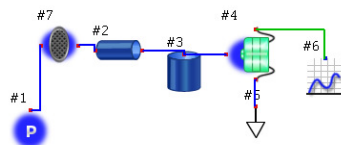
The interface also allows for additional parameter fields to be displayed or hidden, and sorts the list by those parameters.



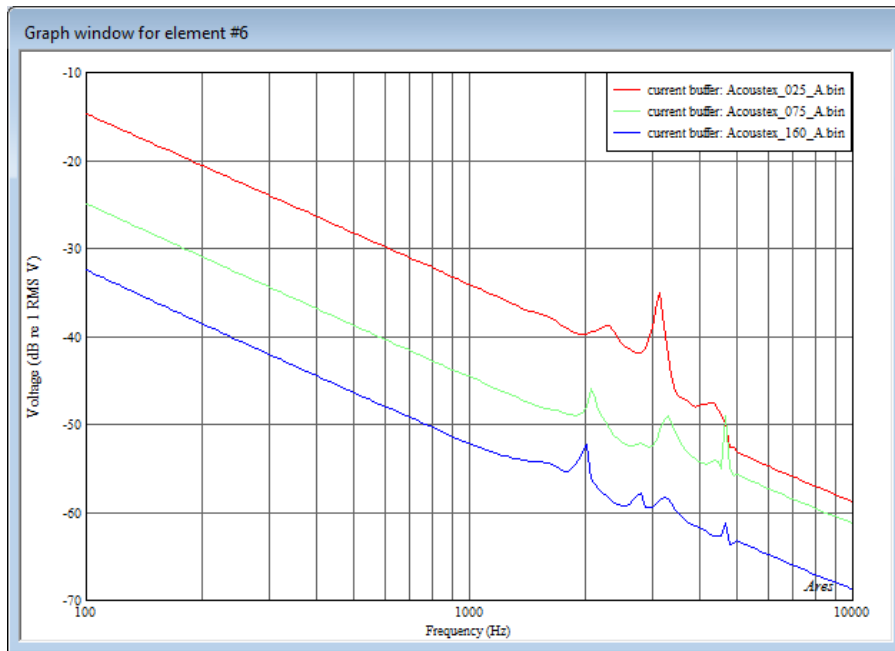
When the *OK* button is pressed, the selected files will be imported into the element. The listbox for the database files will now contain the selected files. To use this impedance data for the element, make sure the *use database impedance data* option is checked.



Below we've placed the *acoustic material* element in front of the microphone port and iterated upon the impedance files that were imported producing the following graph. Note that the effect of the material's different flow impedances generates an overall sensitivity shift. Also, the resonance behavior of the material produces a noticeable effect upon the response between 1 and 5 kHz.



The design challenge is to find materials that can produce a desired frequency response. Without the ability to measure and model the behavior of these materials, you're typically left with trying to minimize the effect of the material. Being able to predict its effect allows you to use the different materials' properties to enhance your design, something which is very hard to do without Ares.



Mechanical Modeling

Ares uses an admittance (or mobility) model representation for mechanical modeling, not an impedance model. The reason for this is that the iconic representation of a mechanical system is largely the same as that used when drawing an admittance model. But for impedance models, parallel mechanical systems become series systems when electrical equivalents are used, and series mechanical systems become parallel electrical equivalents. This is shown below. On the left is a simple mass/spring/damper mechanical system, the middle is the electrical impedance model equivalent representation, and on the right is the Ares admittance model equivalent representation.

mechanical system

impedance model representation

admittance model representation

force ~ voltage

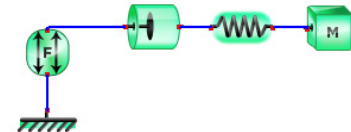
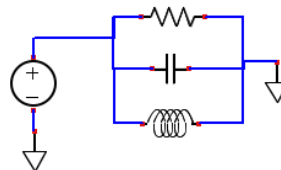
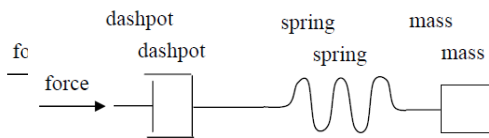
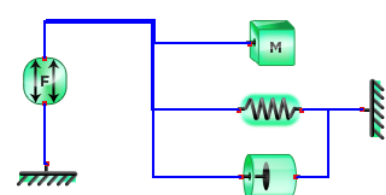
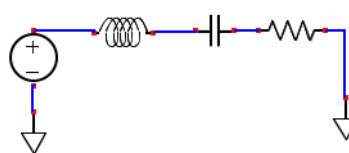
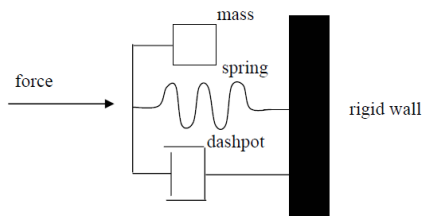
velocity ~ current

node = equal force

force ~ current

velocity ~ voltage

node = equal velocity



It is believed by the author that it's easier for a novice to learn how to build equivalent circuit representations of mechanical systems using an admittance model because the model looks much more like the mechanical system than an impedance model.

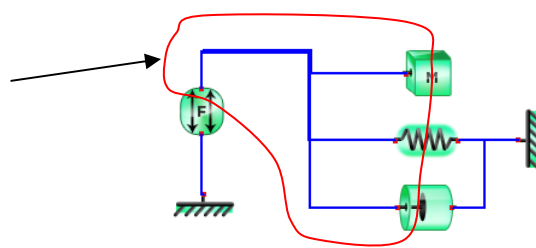
However, building equivalent circuits for mechanical systems can be challenging for the uninitiated. For the experienced user who is used to impedance representations, it can be challenging to change your thinking and generate admittance models.

The author has found that keeping in mind what nodes represent helps tremendously with building the model. For an admittance representation, connecting element nodes together means that all of the connected nodes share the same velocity and the sum of all of the forces into the nodes must equal zero.

So for the model below, all of the nodes that are connected together have the same velocity, where in electrical circuit terms, the velocity is represented by the node voltage.

velocity of all connected nodes are equal

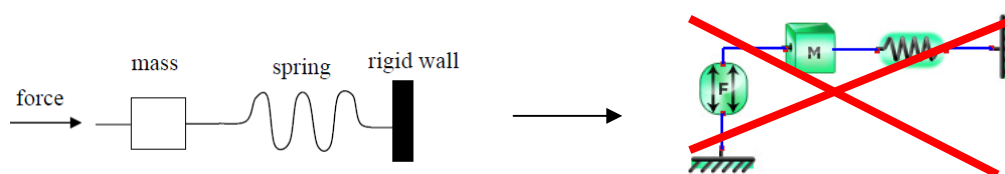
forces into all connected nodes must sum to zero



Also, the forces at all of the connected nodes must sum to zero, where the force is a vector (or directional quantity) with a positive value always being defined to be into the node. In electrical terms, the force is represented by the node current.

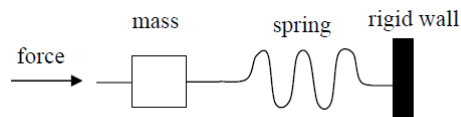
While the mechanical model uses an admittance approach, the graph element only displays impedances (Z), and not admittance (Y), where $Z=1/Y$. Note that the graph element faithfully represents all of the quantities such as velocity, displacement, etc. (The force/velocity values are not reversed as you'd see if you plotted voltage/current using a model with a standard electrical circuit simulator.)

One may confuse the following model as being modeled by a mass and spring in series. This however is not the case.

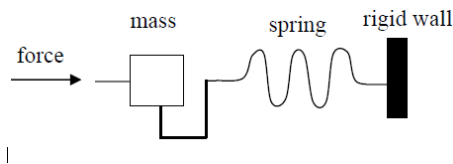


The reason is because a mass is considered to be incompressible, it doesn't matter where you attach the spring to the mass. The spring will have the same motion (displacement or velocity) no matter where you attach the spring to the mass.

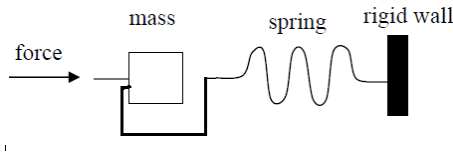
So this representation.



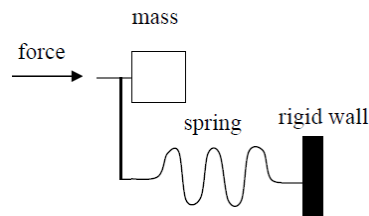
Is the same as this.



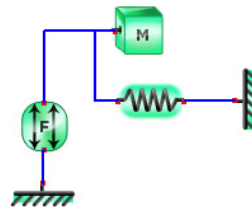
Which is the same as this.



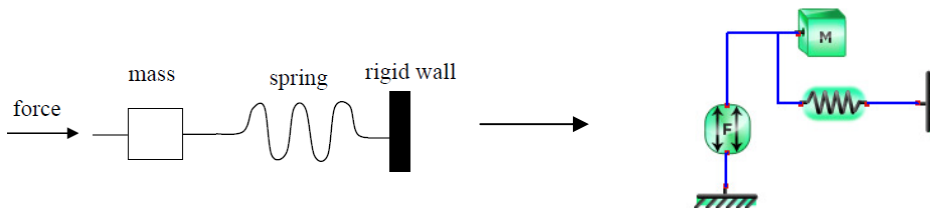
Which is the same as this.



Which in Ares is represented by this model.



So a model which was initially drawn as mechanical elements in series, was actually mechanical elements in parallel, with the confusion coming from where the spring was attached to the mass.



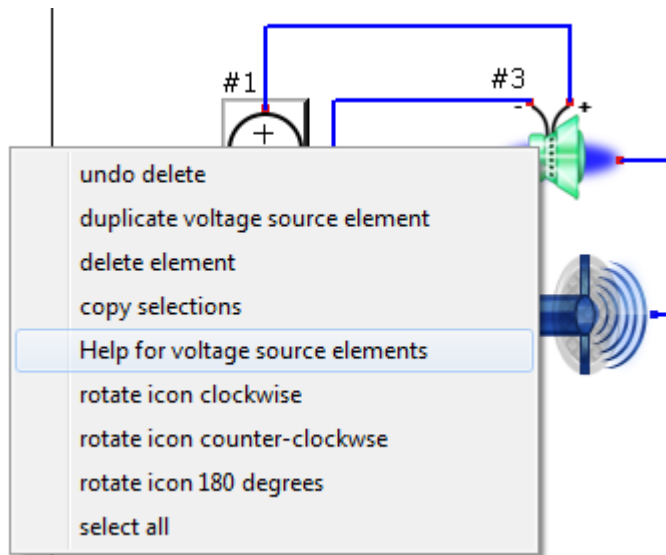
For the users experienced with mechanical impedance modeling, this may at first be confusing. To aid in this, the mass element only has one node which indicates that the entire mass moves together and there is no "velocity drop" across a mass.

Note that the techniques and nuances of equivalent circuit modeling will not be taught in this user's manual. MAE will be supporting them in other material and classes.

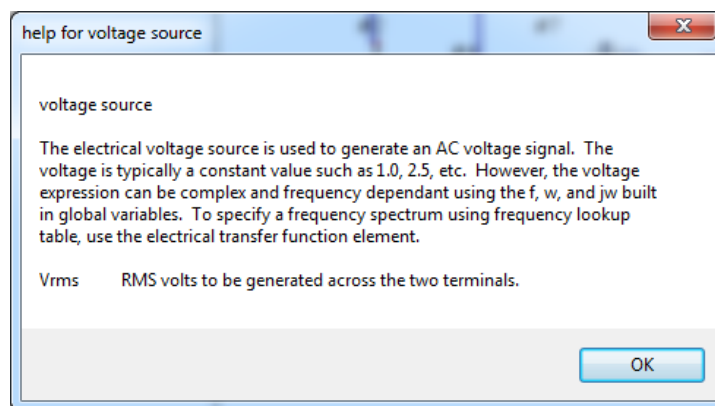
Modeling Elements

This section discusses each element in brief detail. The more complicated elements, such as the Graph element, will be (or have already been) discussed in their own sections.

The text given below is duplicated in the program itself and can be displayed by right clicking on an element and selecting the help entry for that element. For example, the right click menu for the voltage source is shown below.



Selecting Help for voltage source elements brings up the following dialog, displaying the help text.



Electrical Elements



AC voltage source

The standard voltage source is used to generate an AC voltage signal. The voltage is typically a constant value such as 1.0, 2.5, etc. However, the voltage expression can be complex and frequency dependant using the f , w , and jw built in global variables. To specify a frequency spectrum using frequency lookup table, use the electrical transfer function element. Note that this element will have a 0 V value at DC. DC and AC voltages are specified independently to allow for bias voltages to be specified for semiconductor elements.

V_{rms} RMS volts to be generated across the two nodes.



DC voltage source

A DC voltage primarily used to set bias points for semiconductor devices such as MOSFETS used in microphone amplifier models. This will have 0 V at frequencies other than 0 Hz.

V_{dc} DC voltage from the source generated across the two nodes.



absolute AC voltage

The absolute voltage source is used to generate an AC voltage signal. Whereas the standard voltage source specifies a voltage difference between its two nodes, this element specifies a fixed voltage value at a single node. The absolute voltage source is equivalent to a standard voltage source with the negative (-) terminal connected to ground. Note that this element will have a 0 V value at DC. DC and AC voltages are specified independently to allow for bias voltages to be specified for semiconductor elements.

V_{rms} RMS volts at the elements single node.



AC current source

The current source is used to generate an AC current. The current is typically a constant value, but it can be made to have a complex and frequency dependant value by entering a value and using the f , w , and jw built in global variables. (To specify a complex frequency dependent current from a lookup table, use a voltage source, an electrical transfer function and transconductance amplifier elements.) Note that this element will have a 0 A value at DC. DC

and AC voltages are specified independently to allow for bias voltages to be specified for semiconductor elements.

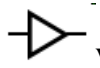
Irms RMS amps from the current source.



DC current source

Provides a DC current. This is useful for providing DC operating points for the nonlinear semiconductor devices such as diodes and mosfets.

Idc DC amps from the current source.



voltage amplifier

Amplifies the input voltage by the specified gain constant K and places the result on the output. The input node (on the left) has an infinite input impedance (i.e. it doesn't load the circuit at all.)

K Amplifier gain. Negative and complex values are allowed.



op amp

An ideal linear operational amplifier. The output is chosen to force the two inputs to be equal. The input impedance into the + - nodes is infinite.



IIR filter

Implements an ideal infinite impulse response filter. Multiple, space separated values can be entered for the Xtaps and Ytaps parameters. If Ytaps is 1, then an FIR filter is realized. If x(n) is the nth Xtaps value, and y(n) is the nth Ytaps value, x(n) is the nth input, and a(n) is the nth output, then

$$a(1)*y(n) = b(1)*x(n) + b(2)*x(n-1) + \dots + b(nb+1)*x(n-nb) - a(2)*y(n-1) - \dots - a(na+1)*y(n-na)$$

Fs The sample rate that the IIR filter taps are evaluated at.

Xtaps A space separated array of X taps that operate on the input values.

Ytaps A space separated array of Y taps that operate on the output values.



electrical transfer function

Transforms the input voltage to the output voltage using a complex, frequency dependent lookup table. The lookup table is provided in the form of text columns and can be loaded from a file or pasted in from the clipboard. The format for the table is specified when data is read in. The user will be presented with choosing between two or three columns, and magnitude or complex formats.

$$V_{out} = V_{in} * H(f)$$

Where V_{out} = voltage out of the element on the right.

V_{in} is the voltage of the left input node.

$H(f)$ is the complex transfer function loaded from the file and linearly interpolated at frequency f .

The electrical transfer function element does not load the circuit that the left input node is connected to. That is, the input impedance of the input node is infinite.

One use for this element is when the input spectrum to a model is not flat. For example, if the response of the system to a P50 speech spectrum is desired, a voltage source can be followed by a transfer function element with a lookup table that specifies a P50 spectrum loaded into it. The magnitude response of the model will now be that for a typical P50 speech spectrum.



electrical-acoustical transfer function

Transforms the input voltage to a pressure using a complex, frequency dependent lookup table. The lookup table is provided in the form of text columns and can be loaded from a file or pasted in from the clipboard. The format for the table is specified when data is read in. The user will be presented with choosing between two or three columns, and magnitude or complex formats.

$$V_{out} = V_{in} * H(f)$$

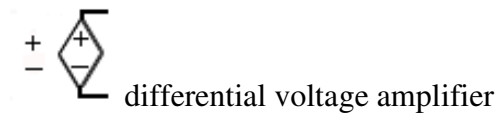
Where V_{out} = voltage out of the element on the right.

V_{in} is the voltage of the left input node.

$H(f)$ is the complex transfer function loaded from the file and linearly interpolated at frequency f .

Units for H is Pa/V.

The electrical-acoustical transfer function element does not load the circuit that the left input node is connected to. That is, the input impedance of the input node is infinite.



Amplifies the voltage difference on the left nodes and produces a voltage on the right. The + - input voltage nodes have infinite impedance.

If V_{in} is the voltage across the left + - input nodes, and V_{out} is the voltage across the output nodes on the right, then

$$V_{out} = G * V_{in}$$

G Amplifier gain.

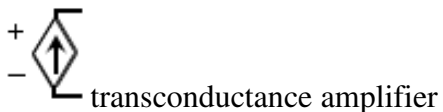


Amplifies the current into the left nodes and produces a current out of the right nodes. There is no voltage drop across the input terminals on the left.

If I_{in} is the current into the top left input node, and I_{out} is the current out of the upper right node, then

$$I_{out} = G * I_{in}$$

G Amplifier gain.



A transconductance amplifier is essentially a voltage controlled current source where the gain is amps/volts, or units of conductance. The + - input voltage nodes have infinite impedance.

If V_{in} is the voltage across the left + - input nodes, and I_{out} is the current out of the upper right node, then

$$I_{out} = G * V_{in}$$

G Amplifier gain.

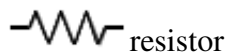


A transresistance amplifier is essentially a current controlled voltage source where the gain is volts/amps, or units of resistance. There is no voltage drop across the input nodes on the left.

If I_{in} is the current into the top left input node, and V_{out} is the voltage across the nodes on the right, then

$$I_{out} = G * V_{in}$$

G Amplifier gain.



A classic electrical resistor.

The 'Load impedance data file' button can be used to load a complex impedance lookup table into the element. Pressing this button will cause a dialog box to appear that will allow you to browse to a file containing the data. The format for a *.txt text file is in three columns:

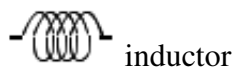
1st column = frequency (Hz)

2nd column = real part of the impedance (electrical ohms)

3rd column = imaginary part of the impedance (electrical ohms)

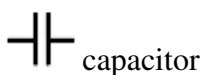
To use this impedance data instead of the resistance specified with R, the 'use file impedance data' must be checked. If the 'use file impedance data option is disabled', then there isn't any impedance data loaded into the element.

R Electrical resistance.



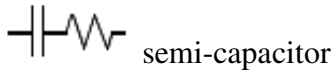
A classic electrical inductor.

L Electrical inductance.



A classic electrical capacitor.

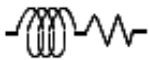
C Electrical capacitance.



semi-capacitor

A semi-capacitance has an impedance of $\sqrt{j/(w*Cs)}$. Note that unlike a real capacitor, the sign on the imaginary part is positive. One use for a semi-capacitor is for implementing discrete models of Knowles Electronics hearing-aid speakers and microphones.

Cs Semi-capacitance.



semi-inductor

A semi-inductance has an impedance of $\sqrt{jw}*Ls$ or $(1+j)*\sqrt{w/2}*Ls$, which consists of a real and imaginary term. Eddy-current losses in a moving coil's pole piece can be effectively modeled by a semi-inductance element in parallel with the coil's inductance element.

Ls Semi-inductance.

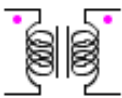


gyrator

A gyrator transforms current to voltage. In this case, the current I_{left} into the upper left terminal generates a voltage $K*I_{left}$ across the terminals on the right, and the current I_{right} into the upper right terminal generates a voltage $-K*I_{right}$ across the terminals on the left. That is

$$\begin{aligned} V_{right} &= K*I_{left} \\ V_{left} &= -K*I_{right} \end{aligned}$$

K Gyrator transformation constant.



transformer

An ideal transformer. If $N1$ is the number of turns on the left and $N2$ is the number of turns on the right, then the turns_ratio will be $N2/N1$. If $I1$ is the current into the left coil (i.e. into the left dot) and $I2$ is the coil into the right coil (i.e. into the right dot), then

$$I1 = I2 * N2/N1.$$

If V1 is the voltage across the left coil and V2 is the coil across the right coil, then

$$V1 = V2 * N1/N2.$$

The violet dots denote classical + terminals of the transformer into which the current flows.

turns_ratio Turns ratio of transformer (N2/N1).



Implements a diode that performs DC signal rectification useful for providing biasing for other semiconductor devices such as a mosfet being used in an amplifier circuit. The only effect that the diode has on the AC analysis is the small signal resistance, rd. When the diode is forward biased, this resistance is $rd = n * Vt / Id$, where $Vt = 26mV$. When the diode is reversed bias, rd is infinite.

Is saturation current or scale current
n ideality factor
Id forward bias DC current through diode
Vd forward bias DC voltage across diode



Implements an N-channel MOSFET to be used as a small signal amplifier. These devices require a DC solution to establish their Ids bias point for the small signal analysis. If the starting frequency for the analysis isn't 0 Hz, the solver will automatically include a 0 Hz DC solution as a part of the solver run. Since the DC characteristics of a MOSFET device is nonlinear, an iterative solution is required. Typically this iterative solver will converge within a few dozen iterations, however it may never converge. If this occurs, you'll need to stop the solver and adjust the circuit to encourage it to converge. Providing a better guess for Isd_initial may be required.

The nonlinear DC model used for the N-channel MOSFET is given below.

for cutoff mode

$$V_{gs} \leq V_{th}$$

$$I_{ds} = 0$$

for linear mode

$$(V_{gs} > V_{th}) \text{ and } (V_{ds} > 0) \text{ and } (V_{ds} \leq (V_{gs} - V_{th}))$$

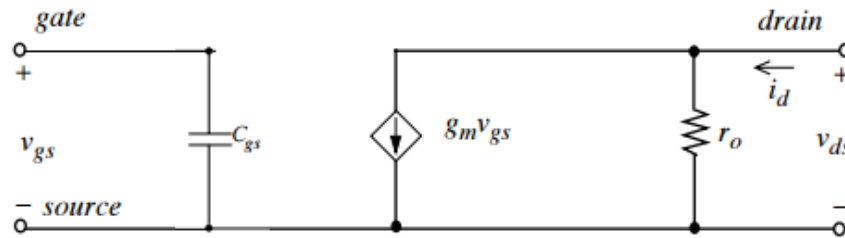
$$I_{ds} = -KP * (W/L) * ((V_{gs} - V_{th}) * V_{ds} - V_{ds} * V_{ds} / 2) * (1 + \lambda * \text{abs}(V_{ds}))$$

for saturation mode

$$(V_{gs} > V_{th}) \text{ and } (V_{ds} > 0) \text{ and } (V_{ds} > (V_{gs} - V_{th}))$$

$$I_{ds} = -(KP/2) * (W/L) * (V_{gs} - V_{th}) * (V_{gs} - V_{th}) * (1 + \lambda * \text{abs}(V_{ds}))$$

Once the DC operating point has been established, the linear small signal model as shown below is applied for all non-zero frequency operating points.



where

$$g_m = 2 * I_{ds} / (V_{gs} - V_{th})$$

$$r_o = 1 / (\lambda * |I_{ds}|)$$

$$C_{gs} = C_{GSO}(\text{farads/meter}) * W(\text{meters})$$

Ids_initial	Initial value to assume for the current into the drain for the nonlinear iterative DC solver. The closer the estimate is, the faster the DC solver will converge. Once a solution has been found for the circuit, this value can be set to the Ids value to make the solver converge almost instantly.
Vth	Threshold voltage.
KP	Transconductance amplification factor for the small signal MOSFET. This is to have units of A/V ² . This parameter is equal to the factor $\mu_n C_{OX}$ seen in the MOSFET equations.
W	Channel width.
L	Channel length.
lambda	Channel-length modulation.
CGSO	Gate-source overlap capacitance per channel width.
Ids	DC drain-source current from DC solver.
Vgs	DC gate-to-source voltage from DC solver.
Vds	DC drain-to-source voltage from DC solver.



Implements a P-channel MOSFET to be used as a small signal amplifier. These devices require a DC solution to establish their Ids bias point for the small signal analysis. If the starting frequency for the analysis isn't 0 Hz, the solver will automatically include a 0 Hz DC solution as a part of the solver run. Since the DC characteristics of a MOSFET device is nonlinear, an iterative solution is required. Typically this iterative solver will converge within a few dozen iterations, however it may never converge. If this occurs, you'll need to stop the solver and adjust the circuit to encourage it to converge. Providing a better guess for Ids_initial may be required.

Except for the threshold voltage, V_{th} , the DC current and voltages are referenced so they are positive quantities for a P-channel MOSFET when the device is being used as an amplifier, the DC current is into the source and out of drain, so its referred to as I_{sd} instead of the more common I_{ds} value referred. Note that $I_{sd} = -I_{ds}$. Same with the V_{sg} and V_{sd} voltages, $V_{sg} = -V_{gs}$ and $V_{sd} = -V_{ds}$.

The nonlinear DC model used for the P-channel MOSFET is given below.

for cutoff mode

$$\begin{aligned} -V_{gs} &\leq -V_{th} \\ I_{ds} &= 0 \end{aligned}$$

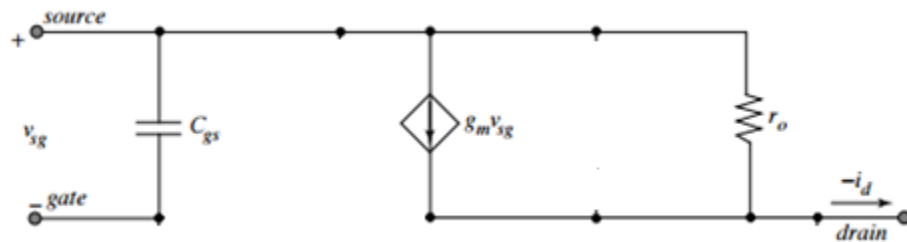
for linear mode

$$\begin{aligned} &(-V_{gs} > -V_{th}) \text{ and } (-V_{ds} > 0) \text{ and } (-V_{ds} \leq (-V_{gs} + V_{th})) \\ I_{ds} &= -KP * (W/L) * ((V_{gs} - V_{th}) * V_{ds} - V_{ds} * V_{ds} / 2) * (1 + \lambda * \text{abs}(V_{ds})) \end{aligned}$$

for saturation mode

$$\begin{aligned} &(-V_{gs} < -V_{th}) \text{ and } (-V_{ds} < 0) \text{ and } (-V_{ds} > (-V_{gs} + V_{th})) \\ I_{ds} &= -(KP/2) * (W/L) * (V_{gs} - V_{th}) * (V_{gs} - V_{th}) * (1 + \lambda * \text{abs}(V_{ds})) \end{aligned}$$

Once the DC operating point as been established, the linear small signal model as shown below is applied for all non-zero frequency operating points.



where

$$\begin{aligned} g_m &= 2 * I_{ds} / (V_{gs} - V_{th}) \\ r_o &= 1 / (\lambda * \text{abs}(I_{ds})) \\ C_{gs} &= C_{GSO}(\text{farads/meter}) * W(\text{meters}) \end{aligned}$$

$I_{sd_initial}$	Initial value to assume for the current into the source for the nonlinear iterative DC solver. The closer the estimate is, the faster the DC solver will converge. Once a solution has been found for the circuit, this value can be set to the I_{sd} value to make the solver converge almost instantly.
V_{th}	Threshold voltage. This is a negative quantity for a P-channel MOSFET.
KP	Transconductance amplification factor for the small signal MOSFET. This is to have units of A/V^2 . This parameter is equal to the factor $\mu_n C_{OX}$ seen in the MOSFET equations.
W	Channel width.
L	Channel length.
lambda	Channel-length modulation.

CGSO	Gate-source overlap capacitance per channel width.
Isd	DC source-drain current from DC solver.
Vsg	DC source-to-gate voltage from DC solver.
Vsd	DC source-to-drain voltage from DC solver.

$\begin{matrix} + \\ + \end{matrix} \Sigma =$ dual input summer

Sums the two electrical signals together. The impedance into the + terminals is infinite.

$\begin{matrix} + \\ + \\ + \end{matrix} \Sigma =$ triple input summer

Sums the three electrical signals together. The impedance into the + terminals is infinite.

$\begin{matrix} + \\ + \\ + \\ + \end{matrix} \Sigma =$ quad input summer

Sums the four electrical signals together. The impedance into the + terminals is infinite.

\downarrow ground

The ground element is used to produce a zero electrical voltage. This is equivalent to using an absolute voltage source and setting the voltage to zero.

Mechanical Elements



force

Produces a mechanical force between the two nodes. Since the mechanical admittance (or mobility) representation is used, force is represented by a "current", and so both connectors must be connected to something in order to produce a force. Typically, the bottom node will be connected to a wall. Not connecting one of the nodes means that there is no force on it. If there's no force on one side of the element, the element can't "push" the other node.

F Force produced between the two nodes



velocity

Provides a velocity between two nodes. Since the mechanical admittance (or mobility) representation is used, velocity is represented by a voltage. For example, connecting two 1 m/s velocity sources in series will produce a total of 2 m/s. A wall element is usually connected to one of the nodes to produce a zero velocity reference point.

As with all other parameters, the velocity can be defined to be complex and frequency dependant using the built in global variables j , w , and jw . For example, since displacement is the integral of velocity and since $1/jw$ is the frequency domain integration operator, a velocity of $1/jw$ corresponds to a displacement source of 1.

V Velocity across the two nodes.



mass

Simulates a mechanical mass. Since masses are incompressible, the velocity of the entire mass is equal. Also, since an admittance (or mobility) model is used for the mechanical models, the velocity is modeled as a "voltage". Since there can't be a "velocity drop" (or voltage drop in the electrical domain) across the mass, there is no need for multiple nodes.

M mass



spring

A mechanical spring. The forces on each side of the spring are equal, so if one side of the spring isn't connected to anything, it won't affect the model.

K spring rate



Mechanical damper or dashpot. Dampers are a source of mechanical loss. As with a spring, the forces on each side of the damper are equal, so if one side of the damper isn't connected to anything, it won't affect the model.

Note: While Ares uses a mechanical admittance (or mobility) model, the damper's parameters takes units of mechanical ohms, which is an impedance quantity. A mechanical ohm is defined to be force over velocity or N/(m/s).

R mechanical resistance of damper



BL coupler

Simulates a circular magnet motor structure for a moving coil loudspeaker. It includes the electrical impedance of the coil as well as the mechanical impedance due to the mass of the coil.

The element has two options.

Use magnetic motor model

and Use M-H demag curve for material

When both of these options are disabled, you enter the BL, resistance, inductance and coil mass directly into the element.

If 'Use magnetic motor model' is enabled, you enter the magnetic and voice coil structural information and the element solves for BL, resistance, inductance and the coil mass using an internal magnetic circuit model.

The element also allows you to load a magnet M-H demagnetization data curve into the element and use this to determine the magnetic operation point for the material. When demagnetization data is loaded, the 'Use M-H demag curve for material' option becomes available. When it's selected, the magnet operation point MAG_Br , MAG_HMmax , MAG_BHoper will be computed from the magnet and gap size information.

Note that in all options, the diameter Do of the area that the coil acts upon is required. This is used for end corrections and to convert from mechanical to acoustical properties. Typically, the BL coupler will be used to drive a membrane element. The BL coupler diameter should be the same as the membrane's diameter.

To specify the H vs. M demagnetization curve, click on the 'Load $H\text{-}\mu*M$ demag curve data' button. A dialog box will appear that will allow you to select an ASCII file with the appropriate data. The file is assumed to have a *.TXT extension. The format of the file is two columns. The first column specifies the applied H field in MKS units of A/m. The second column specifies the residual magnetization material $\mu*M$ in MKS units of A/m.

NOTE: This data comes from the second quadrant of the material's hysteresis curve and so the H values must be negative!

BL	Magnetic field in gap times wire length in gap.
Do	Diameter of area that force is distributed over.
Re	DC resistance of coil.
Le	Inductance of coil.

MAG_ID	Inside diameter of magnet (0 if plug magnet).
MAG_OD	Outside diameter of magnet.
MAG_thickness	Thickness of magnet.
MAG_Br	Magnet remanence. Used if demagnetization file isn't specified. Also referred to as 'residual induction'.
MAG_HM_file	File from which demagnetization curve was loaded from.
MAG_BHmax	The maximum energy product possible from the magnetic material.
MAG_BHoperate	The actual operating energy product from the magnetic material. Magnet prices are typically determined by the volume of the material. The material is being used most efficiently on a volume basis if the BH operating point is at or close to its maximum value.
GAP_ID	Inside diameter of the gap.
GAP_OD	Outside diameter of the gap.
GAP_height	Height of the gap.
GAP_B	Magnetic field in the gap.
motor_efficiency	Relative efficiency of the motor structure. Due to flux leakage, not all of the magnetic field will pass through the gap. This represents the percentage that does. Typically this is evaluated with an FEA magnetic model.
WIRE_gauge	Gauge of the wire using the Brown & Sharpe scale.
WIRE_resistivity	Bulk resistivity of wire material. Common values are copper 1.7241e-008 (ohm-m) aluminum 2.824e-008 (ohm-m)
WIRE_density	Density of wire material. Common values are copper 8890 (kg/m ³) aluminum 2700 (kg/m ³)
WIRE_insulation	Thickness of insulation on wire. Two times this value is added to the gauge diameter to the total wire diameter.
WIRE_layers	Number of layers to wind the coil with. More layers make the coil thicker, but if the coil height is larger than the gap, more layers will place more of the wire in the gap.

WIRE_length	Total length of wire used to achieve desired Re.
COIL_windings	Number of windings per layer around the coil.
COIL_height	Height of the coil. If this is larger than the gap height, BL will be lost.
COIL_diameter	Diameter of the coil.
COIL_thickness	Thickness of the coil
COIL_mass	Mass of the wire in the coil.



mechanical-acoustical transformer

Couples the mechanical and acoustic domains by transforming a mechanical velocity into an acoustic volume velocity. Since acoustic sources are almost always dipoles (e.g. a front and back speaker diaphragm), the acoustic side has two nodes, where the volume velocity into the lower node equals the volume velocity out of the upper node.

If U_a is the acoustic volume velocity out of the acoustic node, then

$$U_a = \pi * (D/2)^2 * V_m$$

where D is the diameter of the acoustic diaphragm being mechanically driven and V_m is the mechanical velocity.

D diameter of acoustic source



mechanical transfer function

Transforms the input velocity to the output velocity using a complex, frequency dependent lookup table. The lookup table is provided in the form of text columns and can be loaded from a file or pasted in from the clipboard. The format for the table is specified when data is read in. The user will be presented with choosing between two or three columns, and magnitude or complex formats.

$$V_{out} = V_{in} * H(f)$$

Where V_{out} = velocity out of the element on the right.

V_{in} is the velocity of the left input node.

$H(f)$ is the complex transfer function loaded from the file and linearly interpolated at frequency f .

The mechanical transfer function element does not load the mechanical structure its input is attached to. . That is, the input impedance of the input node is infinite.



displacement controlled voltage source

Converts a mechanical displacement into a voltage through a conversion constant. This can be useful to model how a microphone converts the displacement of the diaphragm into a voltage. This element does not load the mechanical circuit, and so does not provide any feedback to the mechanical solution.

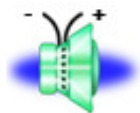
K Conversion constant used to convert displacement to voltage.



wall

A wall represents a zero velocity condition. A wall is typically used for a force element to 'push against'. (Since for an admittance (or mobility) representation, velocity is represented by a voltage, this is would be an electrical ground in an electrical circuit simulator.)

Acoustical Elements



speaker

A moving coil loud speaker. The speaker has two electrical nodes (+ and -) on the top of the icon, and two acoustic nodes which represent the front and rear of the speaker.

The traditional set of parameters for a moving coil speaker is the Thiele-Small (TS) parameters. There isn't a single set of parameters that define TS parameters. The speaker element currently supports two representations. One uses mass/compliance/resistance. The other uses resonant frequency/Q/equivalent air volume.

The 'Load speaker params from file' button allows for speaker parameters to be loaded from a file. Presently three types of files are supported: *.ARES_SPK , *.TXT, and *.KLARES.

*.ARES_SPK files are custom binary files that can hold several types of speaker parameters in the same file including the TS parameters.

For the advanced MAE speaker parameters, more dynamic behavior of the speaker is captured, including internal volume and higher order modal effects. Also, the speaker may only have a

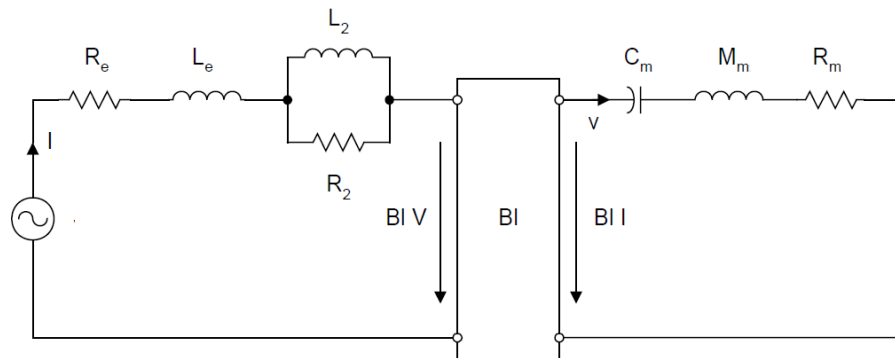
single acoustic port, in which case only the front acoustic port, the one on the right, is used. This is the case for most 'hearing aid' speakers. Also, the advanced MAE speaker parameters can characterize any linear device, including piezo or balanced armature.

*.KLARES files are a text based file generated by the Klippel LPM (Linear Parameter Measurement) speaker parameterization tool. The files look like this:

```
SourceDesc = 'LPM Parameter';           % Source description
NumMaterial = 0;           % Number of material
Sd = 0.049086999;          % [[m^2]] :Surface/diameter of driver
Fmax = 1500;   % [[Hz]] :
Fmin = 60;     % [[Hz]] :
SplitOctave = 16;
StimAmpEff = 0.70710677;    % [[V]] :
StimAmpDBU = -3.8021126;   % [[dB]] :
Mass = nan;    % [] :
Volume = nan;  % [] :
RShunt = 0;    % [[Ohm]] :
UseShunt = 'false';
FixedMmd = nan;    % [[kg]] :
R_E = 5.3661113;    % [[Ohm]] :
L_E_0 = 0.00083420566;    % [[H]] :
L_2 = 0.0011805015;    % [[H]] :
R_2 = 5.4161461;    % [[Ohm]] :
K_LEACH = 0.012161268;
n_LEACH = 0.74551348;
Krm_WRIGHT = 0.00038905611;    % [[Ohm]] :
Erm_WRIGHT = 1.0361816;
Kxm_WRIGHT = 0.0070239928; % [[Ohm]] :
Exm_WRIGHT = 0.8008108;
Cmes = 0.0003233547; % [[F]] :
Lces = 0.042711235;    % [[H]] :
Res = 41.143382;    % [[Ohm]] :
F_S = 42.826194;    % [[Hz]] :
MMS = 0.017091974;    % [[kg]] :
Rms = 1.2847335;    % [[kg/s]] :
C_MS_0 = 0.00080803296;    % [[m/N]] :
Kms = 1237.5733;    % [[N/m]] :
b_0 = 7.2703702;    % [[N/A]] :
LambdaS = 0.028471029;
Qtp = 0.42173854;
Q_MS = 3.5798804;
Qes = 0.46690466;
Qts = 0.41303474;
Mmd = 0.0048026456; % [[kg]] :
Vas = 0.27553837;    % [[m^3]] :
Eta0 = 0.044564061;
Lm = 98.689848;    % [[dB]] :
rmseZ = 0.040846359;
rmseHx = 0.017132032;
B0_imported = nan;    % [[N/A]] :Bl(X=0)
MMS_imported = nan; % [[kg]] :Mms
R_E_imported = nan; % [[Ohm]] :Re(dTv=0)
```

*.TXT files are text files that contain lines that hold classic or Thiele-Small parameters in MKS units. The format is the parameter name, a space and the MKS value for that parameter. An example of the contents of such a file is:

```
Re 30
Le 100e-6
R2 0
L2 0
BL 1
Do 16e-3
Mm 0.1e-3
Cm 2.3e-3
Rm 0.3
Fo 300
Qms 0.7
Vas 12e-6
```



If a parameter isn't included in the *.TXT file, it will be set to zero and reported in a dialog box after loading the parameters.

Radio buttons at the bottom of the parameter dialog box control which set of parameters are used for simulation. Two standard Thiele-Small representations are always available. The advanced MAE speaker parameters will be available if they've been loaded into the element.

Re	DC voice coil resistance of the speaker
Le	Voice coil inductance
R2	Electrical resistance due to eddy current losses. Set to zero if unknown.
L2	Para-inductance of voice coil. The parallel combination of R2 and L2 appears in series with Re and Le. Set to Zero if unknown.
BL	BL product (magnetic field * coil length in the gap)
Do	Diameter of the effective radiating surface area of the diaphragm
Mm	Moving mass of cone and attached air
Cm	Diaphragm compliance
Rm	Mechanical resistance of the diaphragm
Fo	Fundamental resonant frequency of cone
Qms	Q of the diaphragm's resonance
Vas	Equivalent air volume that has same Cm diaphragm compliance
param file	file from which the speaker parameters were loaded



Knowles speaker/receiver

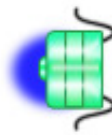
Implements a balanced armature speaker from Knowles Electronics. (These devices are also known as "receivers" in the hearing aid community.) The models used are loaded from the file "VendorComponentLibrary.ARES_VLIB" which should be located in the same directory as the Ares.exe executable. Simply select the desired device from the dropdown listbox that appears in

the element's parameter list. If the device you want isn't listed, contact MAE and request that it be added. The + and - terminals are the electrical inputs, the spout on the other side of the icon is the acoustic output port.



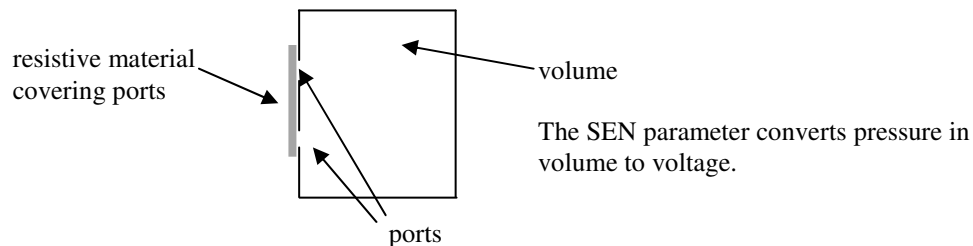
Sonion speaker/receiver

Implements a balanced armature speaker from the Sonion corporation. (These devices are also known as "receivers" in the hearing aid community.) The models used are loaded from the file "VendorComponentLibrary.ARES_VLIB" which should be located in the same directory as the Ares.exe executable. Simply select the desired device from the dropdown listbox that appears in the element's parameter list. If the device you want isn't listed, contact MAE and request that it be added. The + and - terminals are the electrical inputs, the spout on the other side of the icon is the acoustic output port.



omni microphone

An omni (non-directional) microphone. The microphone is modeled using a series of ports backed by a volume. The pressure inside the volume is converted to a voltage that's placed across the top and bottom electrical terminals (the top terminal being +) using a simple sensitivity parameter. (This element does not model an internal diaphragm.) The port/volume structure will load the acoustic circuit, but the electrical output of the microphone is ideal (i.e. it has zero internal source impedance).



Do	Diameter of the port leading into the internal volume.
L	Length of the port leading into the internal volume.
R	Resistivity of the acoustic material covering front ports.
Num	Number of ports leading into the internal volume.
V	Internal volume size.
SEN	Sensitivity of the microphone (applied to internal volume pressure).

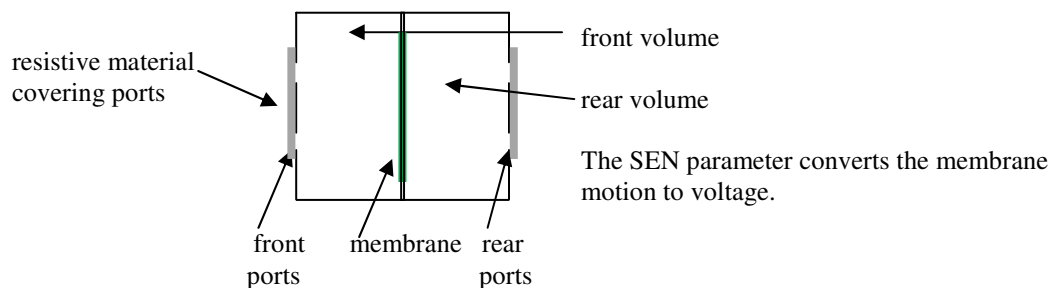


directional microphone

Models a two port pressure gradient directional condenser microphone using fundamental physical parameters. By a judicious choice of the parameters, it can be any type from the family of pressure gradient microphones: cardioid (typically called a 'uni-directional microphone'); bi-directional (typically called 'close talking microphone' or a 'figure 8 pattern'), or a super-cardioid (also referred to as a 'uni-directional microphone').

The default parameters realize a cardioid microphone.

The microphone is modeled as a front damped-port/volume structure, a rear damped-port/volume structure, with a membrane in between. The membrane is characterized by a single resonance mass-compliance-damping structure.



An omni microphone can be realized by essentially closing off the rear ports by making them extremely small, and/or making the rear resistive cover resistance extremely large.

To drive the microphone assembly, a plane wave or point source should be used. For a bare microphone, it is important to get the spacing between the two nodes in the plane wave or point source to be accurate. Typically this will be the length plus the radius of the microphone. Since the response is highly sensitive to this spacing, it may be necessary to experiment with it when realizing a design.

Df	Diameter of front microphone ports into front cavity.
Lf	Length of front ports.
Rf	Resistivity of the acoustic material covering front ports.
Num_f	Number of front microphone ports.
Vf	Volume of front microphone cavity.
Mm	Mass of diaphragm.
Cm	Mechanical compliance of diaphragm.
Rm	Mechanical resistance of diaphragm.
Dm	Moving diameter of diaphragm.
Vr	Volume of rear microphone cavity.
Dr	Diameter of rear microphone ports.
Lr	Length of rear ports.

Rr	Resistivity of the acoustic material covering rear ports.
Num_r	Number of rear microphone ports.
SEN	Free field sensitivity of microphone.
Fsen	Frequency that sensitivity is specified at.
Lmic	Length from front of mic to rear of mic.
Dmic	Diameter of microphone.



Knowles omni microphone

Implements a omni directional microphone Knowles Electronics. The models used are loaded from the file "VendorComponentLibrary.ARES_VLIB" which should be located in the same directory as the Ares.exe executable. Simply select the desired device from the dropdown listbox that appears in the element's parameter list. If the device you want isn't listed, contact MAE and request that it be added. The + and - terminals are the electrical outputs, the connector on the other side of the icon is the acoustic input port.

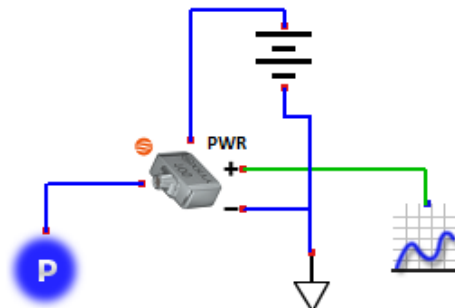


Sonion omni microphone

Implements a omni directional microphone the Sonion corporation. The models used are loaded from the file "VendorComponentLibrary.ARES_VLIB" which should be located in the same directory as the Ares.exe executable. Simply select the desired device from the dropdown listbox that appears in the element's parameter list. If the device you want isn't listed, contact MAE and request that it be added. The + and - terminals are the electrical outputs, the connector on the other side of the icon is the acoustic input port.

The Sonion microphone models require an external DC power supply connected to the top 'PWR' element node. This external power supply MUST be provided. It allows for various power supply voltages to be used, which is useful for determining the current drain from the microphone device.

An example of connecting a Sonion omni microphone is shown below. a simple pressure source is exciting the microphone and the output response is being graphed. The DC voltage source is being used to power the microphone.





displacement controlled voltage

Converts an acoustic particle displacement to a voltage. This is useful for realizing discrete transducer components such as a condenser microphone. The displacement is obtained by dividing the volume velocity by the surface area computed from Do, and then integrating by multiplying by $1/j\omega$.

Note: You must specify the area used to convert volume velocity to particle velocity through the Do parameter. The area is not automatically taken from the nodes that this element attaches to.

Sen Sensitivity of the converter in Volts per displacement.
Do Diameter used to convert the acoustic flow to a displacement (typically the diameter of a membrane that the element is connected to).



acoustical transfer function

Transforms the input pressure to the output pressure using a complex, frequency dependent lookup table. The lookup table is provided in the form of text columns and can be loaded from a file or pasted in from the clipboard. The format for the table is specified when data is read in. The user will be presented with choosing between two or three columns, and magnitude or complex formats.

$$P_{out} = P_{in} * H(f)$$

Where P_{out} = pressure out of the element on the right.

P_{in} is the pressure of the left input node.

$H(f)$ is the complex transfer function loaded from the file and linearly interpolated at frequency f .

The acoustic transfer function element does not load the circuit that the left input node is connected to. That is, the input impedance of the input node is infinite.

One use for this element is when the input spectrum to a model is not flat. For example, if the response of a microphone to a P50 speech spectrum is desired, a pressure source can be followed by a transfer function element with a lookup table that specifies a P50 spectrum loaded into it. The magnitude response of the microphone will now be that of a typical P50 speech spectrum.



pressure

Acoustic pressure at the node. This specifies the exact pressure without any surface area interface, so there is no end correction applied. An equation can be entered in for the pressure, including complex values such as $j\omega$. However, make sure that you're specifying a unit of Pa

and not dB if you're using complex values. To shape the pressure spectrum, an acoustic transfer function element is useful.

P Pressure at node.



volume velocity

Produces a volume velocity through the terminals. A positive value will produce a flow into bottom connector and out the top connector. No interface area is specified, so end corrections are not applied to the nodes of this element.

U Volume velocity produced by the source.



plane wave source

Simulates a plane wave propagating over two sensing points on an infinite plane surface. The wave fronts are perpendicular to the plane surface and so no baffling occurs (which would raise the pressure above that of the incident wave).

This element is typically used for directional microphone modeling, with the two connectors on the element icon representing the pressure at the two sensing points.

For an angle of 0 degrees, the wave moves from the left node to the right node. At 90 degrees, it is from top to bottom and the wave front excites both sensing nodes simultaneously. At 180 degrees it is from right to left, and at 270 degrees it's from bottom to top with the wave fronts again exciting both nodes simultaneously.

The output of the nodes has zero source impedance, so no loading of the impinging wave is produced by connecting elements to the nodes. For this assumption to be accurate, it requires that the impedance into the circuit that the nodes drive is very large, such as with small microphone sensing ports.

P Pressure of the impinging wave.
angle Angle at which the wave is approaching the two points.
Dsep Distance separating the two sensing points.



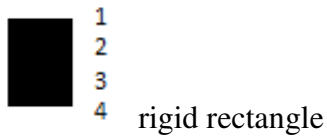
Simulates a spherical wave propagating over two sensing points on an infinite plane surface. The point source for the spherical waves are assumed to be on the plane surface. The strength of the source is specified by entering the desired pressure at the left most node, so a source strength and 2π vs 4π propagation issues are avoided.

This element is typically useful for directional microphone modeling, specifically for bi-directional microphones (also referred to as closing talking or figure eight microphones). The two connectors on the element icon represent the pressure at the two sensing points.

The output of the nodes has zero source impedance, so no loading of the impinging wave is produced by connecting elements to the nodes. For this assumption to be accurate, it requires that the impedance into the circuit that the nodes drive is very large, such as with small microphone sensing ports.

The element doesn't allow for the direction of the point source to be specified. It is assumed that the directional properties of a microphone are to be applied to noise sources much farther away than the point source, so plane wave sources should be used to determine directivity properties.

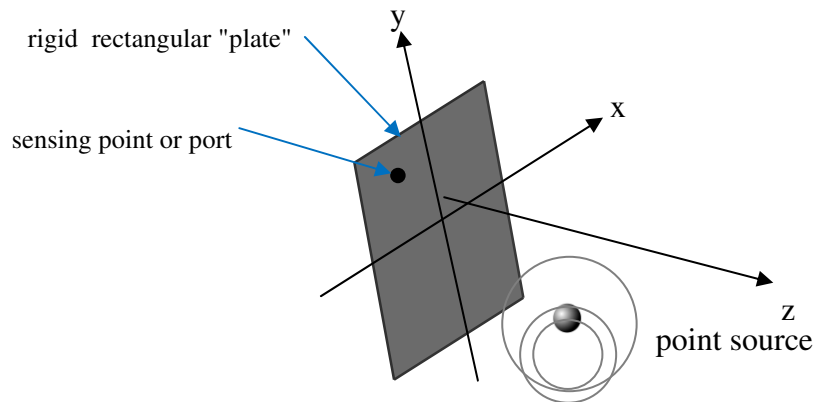
P	Pressure of the impinging wave at the left most point.
R	Radial distance from point source to the center of the two sensing points.
Dsep	Distance separating the two sensing points.



A *rigid rectangle* element simulates the presence of a thin rigid body in space that's being excited by a point source above the rectangle. This largely simulates a talker using a cell phone or a tablet, where the rectangle is the cell phone or tablet, and the point source is the talker. The element allows for multiple sensing points which simulate microphone ports.

The solution is only a numerical approximation and not an exact solution. Keep this in mind when interpreting the results.

The rectangle is in the xy plane at $z=0$. The width is in the x direction and the height in the y direction, with the center of the rectangle at $x=0, y=0$. The xy locations of the sensing ports are specified. They can be either to be on the surface of the rigid rectangle, either front or back surface, where the front is the side where the point source is located.



Being able to predict the microphone pressure on the front and back of a cell phone or tablet is useful in determining microphone locations.

Four sensing locations are presently supported.

width the width of the rigid rectangle in the x axis direction
height the height of the rigid rectangle in the y axis direction
Psrc the pressure of the point source as measured at a distance of 1 inch or 25.4mm. This is the common mouth reference position (MRP) for a head and torso simulator (HATS) artificial mouth.
Xsrc the x coordinate of the point source Psrc
Ysrc the y coordinate of the point source Psrc
Zsrc the z coordinate of the point source Psrc. this needs to be a positive value.
X1,Y1 the x,y location of the first sensing port on the rectangle surface
the front or back location is set by the drop down box below these values
X2,Y2, etc. repeat of X1,Y1 but for the other sending port locations.



acoustical membrane

The acoustical membrane element models the mass-compliance-damping (M-C-R) behavior of membranes or plates. The M-C-R values can be entered directly, or computed for the first mode from fundamental principles for a plate or membrane. Plates differ from membranes by the source of the compliance or restoring spring force. For plates, the restoring force comes from the natural structural stiffness of bending the material. For a membrane, the restoring force comes from tension in the stretched material.

If the impedance of a membrane is measured using the Flow Impedance Module, its impedance should be loaded into the acoustical resistance element.

M	Mechanical moving mass of the membrane. Will be computed for the 1st mode if plate or membrane eqs option is selected.
C	Mechanical compliance of the membrane. Will be computed for the 1st mode if plate or membrane eqs option is selected.
R	Mechanical resistance of the membrane. This represents the damping or energy loss for the membrane or plate. This will not be computed from the plate or membrane equations, but must be manually entered. A value of 0 will represent an undamped membrane resonance. Typically you'll either measure R or estimate its value by selecting a value that provides good fit at resonance between model and a system response measurement.
D	Diameter of the membrane or plate.
Num	Number of membranes or plates in parallel.
h	Thickness of the membrane or plate material. Used in the fundamental equations for computing the plate or membrane's 1st mode frequency.
rho	Density of the membrane or plate material. Used in the fundamental equations for computing the plate or membrane's 1st mode frequency.
E	Young's modulus of the plate material. Used in the fundamental equations for computing the plate's 1st mode frequency.
v	Poisson's ratio of the plate material. Used in the fundamental equations for computing the plate's 1st mode frequency.
T	Tension of the membrane material. Used in the fundamental equations for computing the membrane's 1st mode frequency.
Fo	Resonant frequency of the membrane or plate's first mode. Computed from M and C.



acoustical material

An element representing a lossy acoustic material (such as a fabric or felt) that can be characterized by a flow resistance or a complex frequency dependent impedance. The flow resistance is provided in units of Rayls, which is normalized by the surface area. Thus, the area of the material that has sound passing through it must be specified. This is done by specifying the diameter of a circle with the equivalent surface area. (The Num parameter allows you to specify multiple circular areas so that the total area will be Num times the individual circular area defined by Do.)

Note that the round port and slotted port elements also have a damping material flow resistance term. If a material is used that has a resistance (or impedance) that can be described by a simple DC flow resistance value (or by equations with complex parts that can be entered into the parameter field), then it is typically easier to use the resistance parameters in the port elements than to use a separate acoustic material element.

The element offers the ability to load impedance files from *.ARES_ZF or *.TXT files, or from the online materials database. The *.ARES_ZF files are produced by the Flow Impedance Measurement Module.

The *.TXT files are text files with three space delimited columns.

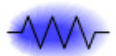
1st column = frequency (Hz)

2nd column = real part of impedance (MKS Rayls, Pa-s/m)

3rd column = imaginary part of (MKS Rayls, Pa-s/m)

The online materials database has already been discussed in its own section.

R	Static flow resistance of the damping material in MKS Rayls. [1 MKS Rayls = pressure drop (Pa) / velocity through material (m/s)]
Do	Diameter of a circle which has the same area of the acoustic material. Note that only the area of the material that sound passes through is to be used in the area calculation.
Num	The number of such Do areas through which sound passes through the material. This can be used to indicate that the material covers an area of Num small holes of diameter Do.



resistor

An acoustical resistor specifies the acoustical resistance between two nodes. Acoustical resistance is the ratio of pressure drop over volume velocity [$\text{Pa}/(\text{m}^3/\text{s}) = \text{acoustical ohms}$]. To compute end corrections, the opening areas for the left and right nodes must be known.

The element offers the ability to load impedance files from *.ARES_ZF or *.TXT files, or from the online materials database. The *.ARES_ZF files are produced by the Flow Impedance Measurement Module.

The *.TXT files are text files with three space delimited columns.

1st column = frequency (Hz)

2nd column = real part of impedance (Acoustic Ohms, Pa-s/m³)

3rd column = imaginary part of (Acoustic Ohms, Pa-s/m³)

The online materials database has already been discussed in its own section.

Acoustical resistor differs from acoustical material in that the acoustical resistor uses acoustical ohms whereas the acoustical material uses MKS Rayls. The fundamental internal unit of impedance that Ares uses is acoustical ohms (pressure over volume velocity). Because of this, acoustical ohms values imported into the acoustical resistor element are treated as independent of area. However, resistance (or impedance) in units of MKS Rayls must be scaled by dividing it by the surface area of the material to achieve units of acoustical ohms.

So the impedance in acoustical ohms that you specify with R, or import into the acoustical resistor element from an impedance file is what's directly applied to the model. However, the impedance specified (or imported into) the acoustical material element is always scaled by the effective area specified by the element's Do and Num parameters.

R	Resistance (or impedance) of air flow through resistor. [Acoustical ohms = pressure/volume velocity = Pa/(m ³ /s) = kg/(m ⁴ s)]
Dleft	Diameter of the equivalent area of the left node.
Dright	Diameter of the equivalent area of the right node.
Num	The number of resistors in parallel.



An ideal compliant air volume.

The compliance is only a function of the total volume of the element, so any combination of diameter and height that achieves the same volume will have the same compliance. However, if the input area of the volume (as defined by its diameter) is smaller than any other element attached to it, an end correction will be applied and a small mass reactance into the element will be realized.

An ideal volume differs from a circular port of the same diameter and length. The circular port's impedance will take into account standing waves along the port, but the volume does not.

By default, the user specifies the height and diameter of a circular volume and displays the resulting volume. The listbox in the element's parameters allows the user to alternatively specify the volume and calculate the height or diameter.

D	The diameter of the volume used for end correction calculations Total volume will be $\pi \cdot H \cdot (D_o/2)^2$.
H	The height of the volume.
Vol	The volume of the cylindrical volume.



The radiation element is used to model the radiation of sound from a hole into an infinite air space. Examples of this are:

- speaker radiating sound from a box into free space
- a port radiating sound into free space
- a leak radiating sound into free space

When the diameter of the baffle is very large, this element essentially realizes the classic baffled radiating source. If the size of the baffle is set equal to the radiating hole diameter, a non-baffled radiating source solution will be used. For finite baffle sizes, a solution between the two will be used.

The blue node on the left side of the element is a listening node. It represents the axial pressure at the specified distance from the radiating point. This port cannot be loaded down, but will always generate the same pressure no matter what elements you attach to it.

Each radiation element radiates into its own mathematical free space and multiple radiating elements do not interact with each other. To see the effect of multiple radiating sources, you must use a summation element to add their pressures together.

A radiating element is frequently attached to the end of a port. For this reason, there are hybrid circular and rectangular ports with a radiating element attached to them. These behave identically to connecting a port and radiating element. Using these hybrid 'radiating port' elements can help simplify the model.

It is important to accurately estimate the value of Dbaff. By default, Dbaff is extremely large and so a baffled solution is realized. But most devices will have a finite baffle size. Leaving Dbaff extremely large will result in a prediction that's up to 6 dB higher than what will be measured. The 6 dB value is the difference between radiating sound into a half space as opposed to a full space, i.e. 2π vs 4π space respectively.

Rdist	Radial distance that the radiated pressure will be calculated for.
Do	Diameter of the Num holes from which the sound is radiating from. (If the holes are non-circular, this should represent the effective area of the individual holes.)
Dbaff	Effective baffle diameter of the surface that the holes are radiating from.



circular port

A circular port (or vent), that comprises a narrow constriction through which sound passes. For low frequencies, this will behave as a simple mass reactance. However the port is implemented as a one dimensional standing wave tube with viscous and thermal effects accounted for, so high frequency standing wave effects will be handled properly. Also, for very small diameters, the viscous drag will introduce the proper lossy behavior for a small tube.

If a port isn't perfectly round, the port's diameter D_o should be specified so the cross section area of the actual port is the same as the circular port.

If the port is close to a circle, the circular port should be used. If the port is closer to a rectangle, the rectangular port should be used. The major difference between the circular and rectangular ports is how the viscous and thermal effects are accounted for.

A resistive damping material is also incorporated into the port on the 'right' side of the element. As the total resistance of the material is scaled according to the port's cross-section area, this is a more convenient way to deal with resistive cloths than using the damping material element.

By default, the user specifies the length and diameter of the port and the element computes the volume, surface area, and mass inertance Ma from those.

When the port length is much shorter than a wavelength, the port's behavior is well captured by its mass reactance, Ma . If the model is performing well with a specific Ma value, you can use the list box to specify an Ma value and compute the length or diameter that provides the same Ma value.

A port can also be used to model a volume, driven at one end. This will accurately model the standing wave along the length of the volume.

L	Total physical length of the port (Ares will calculate the appropriate end correction). The length is in the direction of fluid flow.
D_o	Diameter of the port. If the port is not circular, this should represent an equivalent cross section area of the port.
R	Resistance of the acoustic material placed at the right end of the port. The units are MKS Rayls, which is an area normalized flow impedance. [1 MKS Rayls = pressure drop (Pa) / velocity through material (m/s)]
Num	The number of ports of dimensions L· D_o terminated with resistance R acting in parallel.
Vol	The volume of the port.
S	Cross section area of the port.
Ma	Mass inertance of the port.



tapered circular port

This is a circular port whose radius has a linear taper of Dleft on the left and Dright on the right. See the description for a circular port for more information about ports. Note that the 'right' end always has the larger opening, regardless of how the icon is rotated.

As with the circular port, the listbox allow for the mass inertance (Ma) to be specified and the diameters or length computed to achieve the Ma value.

L	Length of the port.
Dleft	Diameter of the left end of the port.
Dright	Diameter of the right end of the port.
R	Resistance of the acoustic material placed at the right end of the port, so the effective resistance will be scaled according to the area computed from Dright. The units are MKS Rayls, which is an area normalized flow impedance. [1 MKS Rayls = pressure drop (Pa) / velocity through material (m/s)]
Num	The number of ports acting in parallel
Sleft	Cross section surface area of left port computed from Dleft.
Sright	Cross section surface area of right port computed from Dright.
Ma	Mass inertance of the port.



radiating circular port

Combines a circular port and a radiation element. The result is the same as if a radiating element was connected to a circular port element. Since most radiating holes are driven by a port, this element is meant to eliminate the task of creating two elements and matching their areas. See the help for the two elements for more detail.

L	Total physical length of the port.
Do	Diameter of the port. If the port is not circular, this should represent the cross section area of the port.
R	Resistance of the acoustic material placed at the right end of the port where the radiation element is located. The units are MKS Rayls, which is an area normalized flow impedance. [1 MKS Rayls = pressure drop (Pa) / velocity through material (m/s)]
Ro	Radial distance that the radiated pressure will be calculated for.
Dbaff	Diameter of the surface that the holes are radiating from.
Num	The number of ports acting in parallel.
Vol	The inner volume of the port.
S	Cross section area of the port.
Ma	Mass inertance of the port.



rectangular port

A port (or vent), that comprises a narrow rectangular constriction through which sound passes. For low frequencies, this will behave as a simple mass reactance. However, the port is implemented as a one dimensional standing wave tube with viscous and thermal effects accounted for, so high frequency standing wave effects will be handled properly. Also, for very small dimensions, the viscous drag will introduce the proper lossy behavior for a small tube.

The rectangular port is different from the circular port in that the port has a rectangular cross section to simulate a slot. The significance of this is in how viscous drag is taken into account.

A resistive damping material is also incorporated into the port on the 'right' side of the element. As the total resistance of the material is scaled according to the port's cross-sectional area, this is a more convenient way to deal with resistive cloths than using the damping material element.

When the port length is much shorter than a wavelength, the port's behavior is well captured by its mass reactance, Ma . If the model is performing well with a specific Ma value, you can use the list box to specify an Ma value and compute the length or diameter that provides the same Ma value.

A port can also be used to model a volume, driven at one end. This will accurately model the standing wave along the length of the volume.

L	Total physical length of the port.
W	Width of the port, which should be greater than its height, i.e. $W > H$.
H	Height of the port, which should be smaller than its width, i.e. $H < W$.
R	Resistance of the acoustic material placed at the right end of the port. The units are MKS Rayls, which is an area normalized flow impedance. [1 MKS Rayls = pressure drop (Pa) / velocity through material (m/s)]
Num	The number of ports acting in parallel.
Vol	The inner volume of the port.
S	Cross section area of the port
Ma	Mass inertance of the port.



tapered rectangular port

A rectangular port whose height has a linear taper of H_{left} on the left and H_{right} on the right. See the description for regular slot port for more information. Note that the 'right' end always has the larger opening, regardless of how the icon is rotated.

L	Total physical length of the port (Ares will calculate the appropriate end correction. The length is in the direction of fluid flow.
W	Width of the port, which should generally be greater than its height, i.e. $W > H$.
H_{left}	Height of the port on the left, which should be smaller than its width, i.e. $H < W$.
H_{right}	Height of the port on the right, which should be smaller than its width, i.e. $H < W$.
R	Resistance of the acoustic material placed at the right end of the port. The units are MKS Rayls, which is an area normalized flow impedance. [1 MKS Rayls = pressure drop (Pa) / velocity through material (m/s)]
Num	The number of ports of dimensions acting in parallel.
Vol	The inner volume of the port.
S_{left}	Cross section area of the port on the left.
S_{right}	Cross section area of the port on the left.



radiating rectangular port

Combines a rectangular port and a radiation element. The result is the same as if a radiating element was connected to a rectangular port element. Since most radiating holes or slots are driven by a port, this element is meant to eliminate the task of creating two elements and matching their areas. See the help for the two elements for more detail.

L	Total physical length of the port.
W	Width of the port, which should be greater than its height, i.e. $W > H$.
H	Height of the port, which should be smaller than its width, i.e. $H < W$.
R	Resistance of the acoustic material placed at the right end of the port, where the radiation element is located. The units are MKS Rayls, which is an area normalized flow impedance. [1 MKS Rayls = pressure drop (Pa) / velocity through material (m/s)].
Rdist	Radial distance that the radiated pressure will be calculated for.
Dbaff	Diameter of the surface that the holes are radiating from.
Num	The number of ports of dimensions $L \cdot W \cdot H$ terminated with resistance R acting in parallel.
Vol	The inner volume of the port.
S	Cross section area of the port
Ma	Mass inertance of the port.



exponential horn

Models an exponential horn whose cross sectional area varies exponentially according to $S(x) = p (Dl/2)^2 * \exp(mx)$ where m is the flair constant and x varies from 0 to L . Note that the right side of the horn (with the largest cross section area) does not simulate any radiation conditions. To model a radiating horn, you must connect the right side to a radiation element.

Also note that by choosing the appropriate port parameters, parabolic horns and conical horns can be modeled using the tapered rectangular port and tapered circular ports, respectively.

L	Length of the port .
Dleft	Diameter of the left end of the port.
Dright	Diameter of the right end of the port.
R	Resistance of the acoustic material placed at the right end of the port. The units are MKS Rayls, which is an area normalized flow impedance. [1 MKS Rayls = pressure drop (Pa) / velocity through material (m/s)]
Num	The number of horns in parallel.
m	Flair constant of horn $S(x) = p (Dl/2)^2 * \exp(mx)$ where x goes from 0 to L .
Fc	Cutoff frequency (no sound propagation below this frequency).



ear simulator

Represents the acoustical loading of artificial ear couplers. The red node on the left side of the element icon represents the load of the ear and specifies the ear reference pressure (ERP). The blue sensing node on the right side of the element icon represents the pressure that would be measured by the ear simulator's microphone assuming a simple scalar calibration provided by the ear simulator.

The input of these ear simulators are taken to be ERP. The response of these ear simulators are not flat to a constant pressure at ERP, however their response at 500 or 1000 Hz is generally taken as their scalar sensitivity. The response at the other frequencies is measured and provided as a curve with the simulators as a correction function.

The ear simulator element uses a typical sensitivity function to convert the ERP pressure to a pressure measured by the ear simulator assuming a simple scalar sensitivity value. So assume:

Perp = pressure at ERP in dB SPL

Vmic = voltage measured by the microphone in dBV

Sen = scalar sensitivity of the microphone at 500 or 1000 Hz in dBV@1Pa

Corr = frequency dependent correction function

Then

$$V_{mic} = P_{erp} - 94 + Sen + Corr$$

Where 94 is the conversion from 1 dBPa to SPL.

It is common for acoustic measurement systems to apply a scalar sensitivity to a microphone response and report the SPL from the microphone. When this is done for an ear simulator, the pressure reported is

$$P_{meas} = V_{mic} - Sen + 94$$

From the previous equation, we see that this is

$$P_{meas} = P_{erp} + Corr$$

Pmeas is the pressure that the ear simulator provides when a graph element is connect to the right side node.

There is a third node on the lower left corner of the ear simulator. The pressure shown when the graph is connected to this node is the equivalent diffuse field response that the ear simulator's ERP pressure represents for a HATS in a diffuse field. Generally, most audio equipment (such

as headphones and speakers) are designed to have a flat diffuse field response. However, telephony equipment is generally designed so the ERP response fits inside of a particular mask.

The listbox chooses the type of ear simulator to use in the model. The checkboxes select which ear simulators to iterate upon when the iteration button to the right of the listbox is pressed.

Four types of ear couplers are supported:

B&K 4185 or ITU Type 1

The ITU Type 1 simulates a condition where there is a perfect seal between the phone and the ear.

B&K 4195 low leak & high leak or ITU Type 3.2

The ITU type 3.2 simulates a condition where there is a small and large leak between the phone and the ear. Internally these are given the numeric type '4195' and '4195.1' respectively.

IEC-711 or ITU Type 2

The IEC-711 or ITU Type 2 looks like a large microphone cartridge, but it actually contains acoustical chambers that give it an acoustical impedance that matches a part of the ear canal and ear drum. It is commonly used in HATS artificial ears.

B&K 4128C or ITU Type 3.3

The ITU Type 3.3 is a rubberized ear pinna on a HATS that uses an IEC-711 to perform more realistic listening measurements than are obtainable with the Type 1 or Type 3.2 simulators. Unfortunately, the acoustical load that a phone will experience when placed against such a rubberized ear depends on placement, orientation, the force that is exerted against the ear, and the shape of the physical phone. This makes it impossible to define an independent impedance into such an ear. However, the impedance into the ear was measured for different loading forces and is used by the simulator.



dual input summer

Sums two acoustical signals together. The impedance into the summation terminals is infinite.



triple input summer

Sums three acoustical signals together. The impedance into the summation terminals is infinite.



quad input summer

Sums four acoustical signals together. The impedance into the summation terminals is infinite.



pressure release

Forces a zero acoustic pressure condition at its node. (The terms 'pressure release' comes from underwater acoustics where the water-air boundary is considered to be a pressure release boundary due to the substantially reduced characteristic impedance of air as compared to water.)

Thermal Elements

Ares presently offers very limited thermal modeling capabilities. If more features are desired, please contact MAE as we're constantly looking to broaden our tools and the problems they can solve.



heat source

Provides a constant supply of heat into a thermal circuit. For example, if 3 amps are being dissipated into a 4 ohm resistor, the total power is $I^2 \cdot R = (3)^2 \cdot 4 = 36$ watts.

W The amount of thermal power coming out of the source.



constant temperature

A fixed temperature. This can simulate an infinite heatsink at the specified temperature, where no matter how much heat goes into it, the temperature never rises.

T Temperature of the node.



thermal mass

A thermal mass whose temperature rises and falls with the amount of heat energy it absorbs or gives off. The thermal mass is a combination of the volume of the mass, its density, and heat capacity.

V Volume of the mass.
rho Density of the material.
h Heat capacity of the material.



thermal resistor

A thermal resistance. The heat flow through the material is 'resisted' by the value entered. A large value would be a thermal insulator, a small value would allow the heat to readily flow across the element's nodes.

R Thermal resistance.

Miscellaneous Elements



graph

The graph element is the only way to see the results from the model's simulation. It plots the solution values at the node that the graph element is connected to. The graph element can only be connected to one node at a time. The node data is copied to the graph's 'connector node' buffer. The data in the connector node buffer can be copied to other buffers by pressing the buffer buttons, which are labeled A-Z.

The graph element has many features that have already been discussed earlier in the manual.



notes

Allows you to enter text for documenting the model.

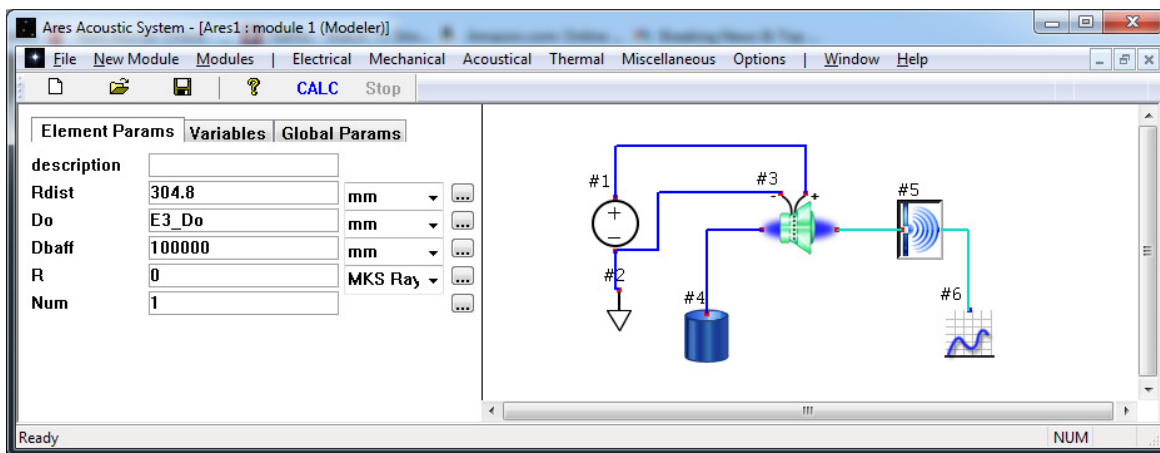
Examples

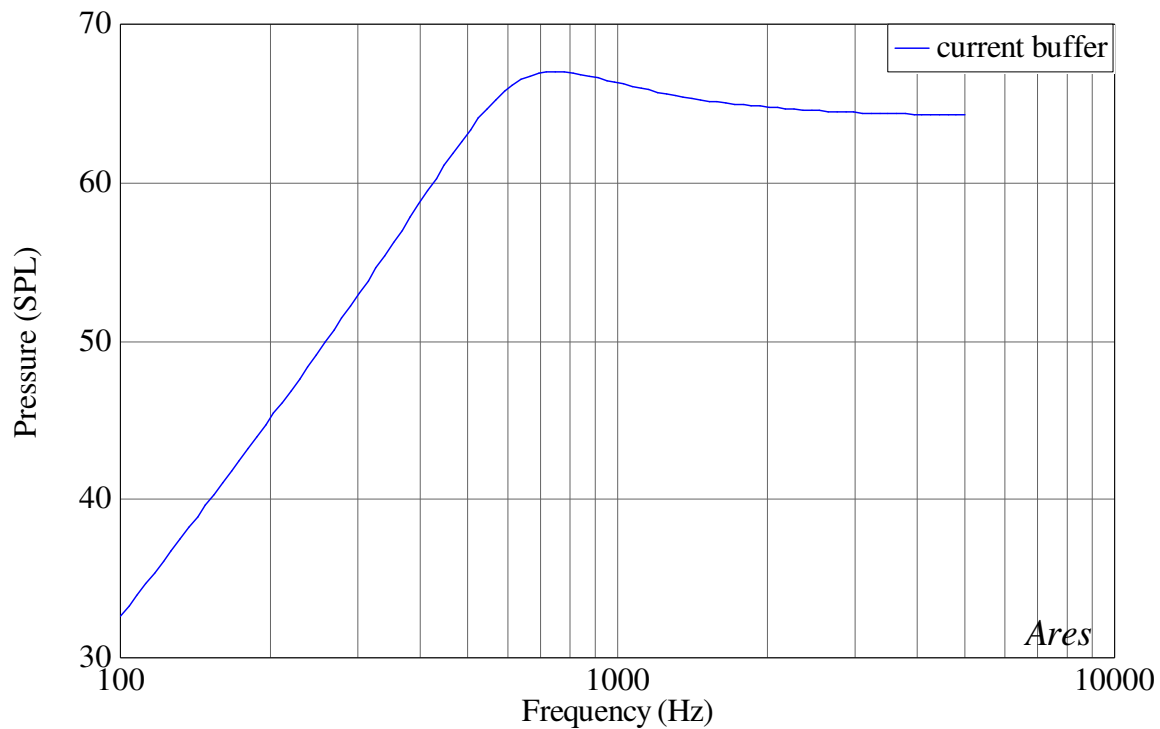
This section provides multiple examples of using Ares to model acoustical systems. These are meant to be simple examples that can be built upon to model more complicated systems. MAE also offers training classes to help further advance the user's skills.

Unless otherwise noted, the default parameter values will be used.

Example 1 Bass Reflex Speakerphone

A bass reflex speaker boots the low frequency output by adding a resonant port to the rear volume. Before we add the resonant port, we'll just model what the speaker does without it. Build the model below using the default parameters. The only parameter we'll change is the diameter of the radiating element, #5. Change its Do parameter to E3_Do, which means "use the Do value from element #3". (Note that you can only use parameters from elements with a lower index #, so element #5 can only use parameters from element #'s 1-4.)

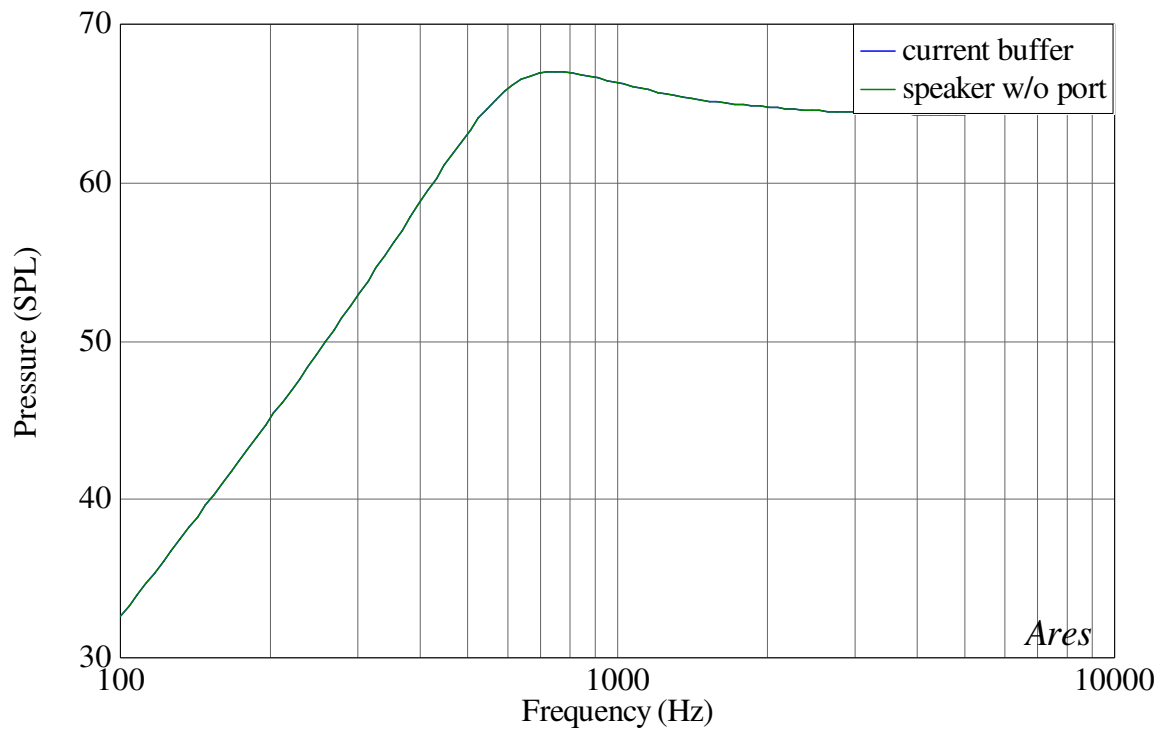




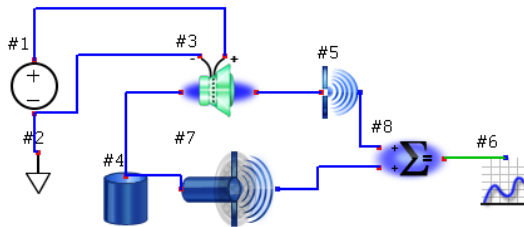
Hit the CALC button (or the F5 key) to generate the graph above. This is the response of the speaker without a resonant port. Let's save this curve for later reference. Select the graph element, press the "A" button to copy the data into buffer A, select the checkbox to the left of the "A" button, and type "speaker w/o port" into the buffer's description. The graph is now as shown below.

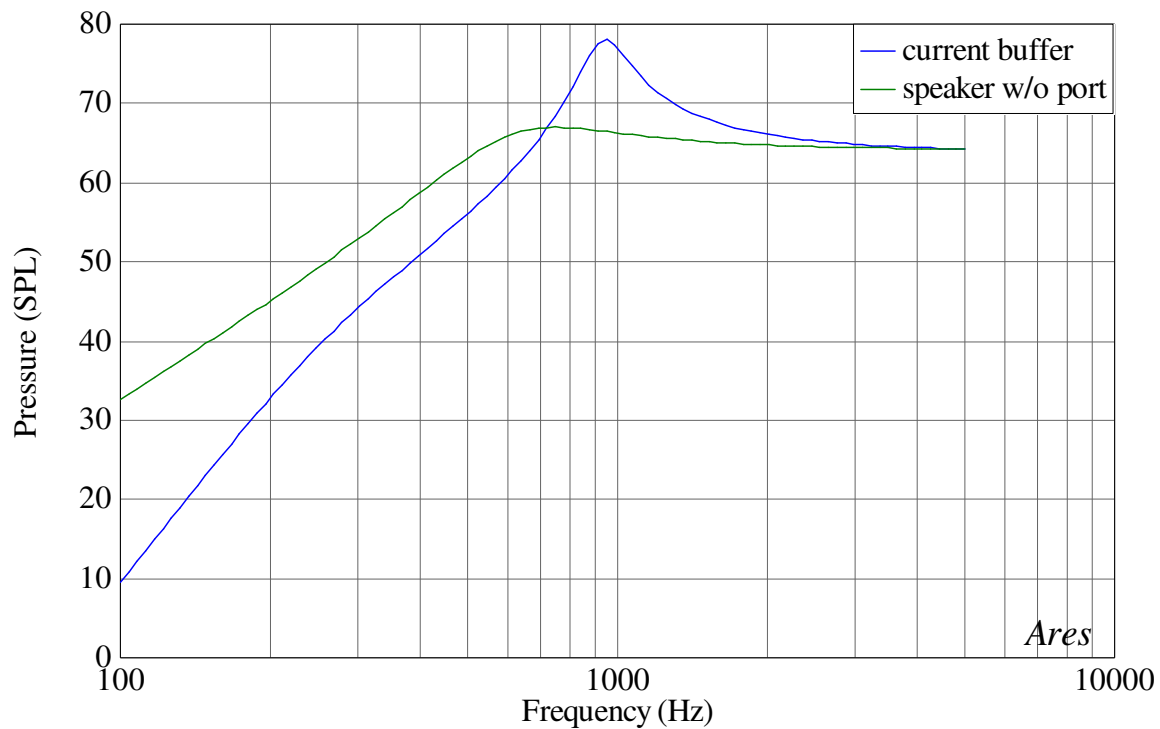
Plotting buffers

<input checked="" type="checkbox"/>	A	speaker w/o port
-------------------------------------	----------	------------------

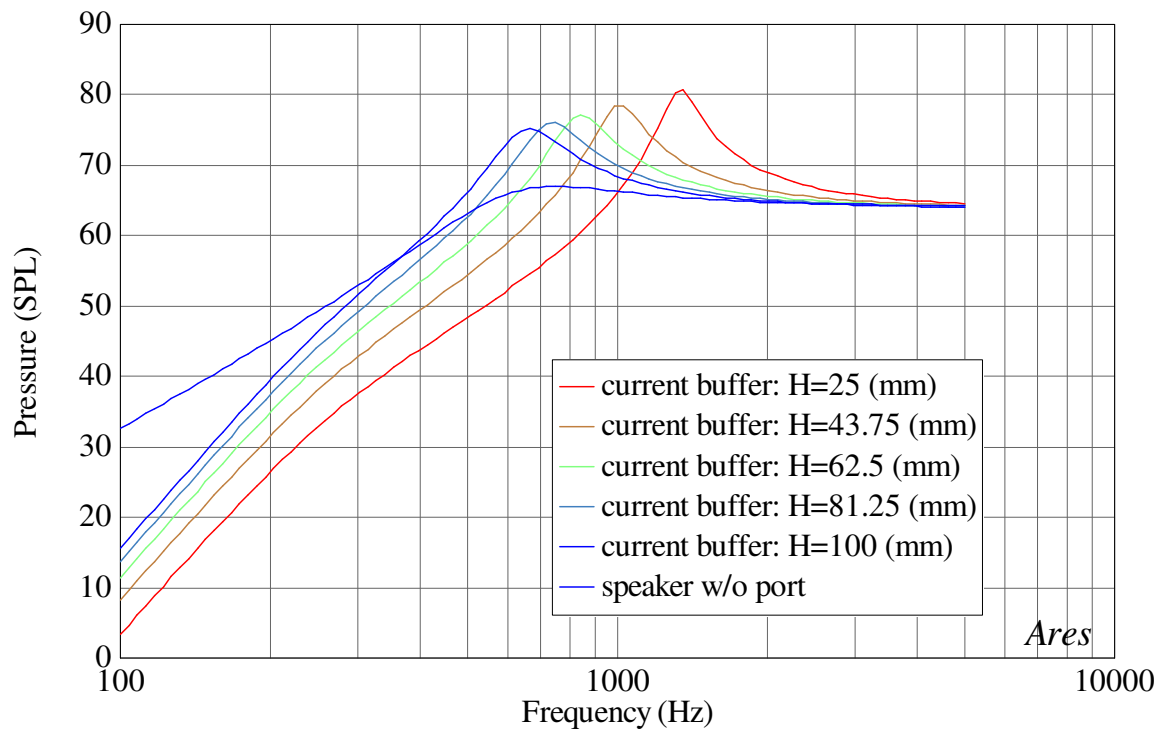


Add a radiating circular port and a dual summer to the model as shown below. Pressing the CACL button (or F5 key) will generate the following graph. Note that the output at 950 Hz has been significantly increased, but at the cost of less output below 700 Hz. This is a typical tradeoff for a bass reflex speaker design.



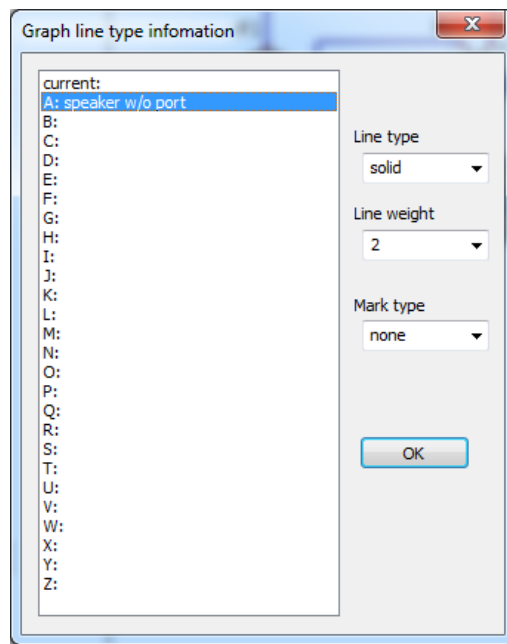


To help optimize the choice of the rear volume and port size, select the volume element #4 and iterate on its height to produce the following graph.

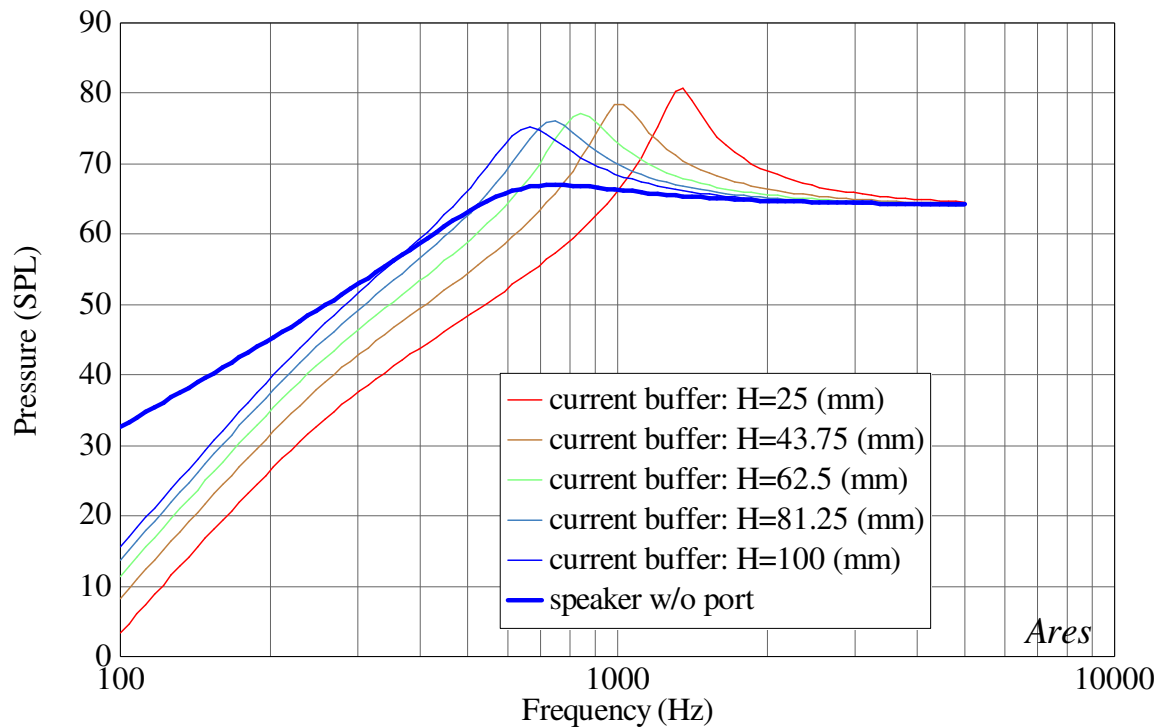


Now you can choose a rear volume height that best delivers the response you desire.

Note that the data in buffer A is hard to see as its color is blending in with the other iteration curves. We can make it stand out by using the *Set buffer line* button in the graph parameters to bring up the following dialog box, and change the line weight for buffer A to 2. Our baseline curve in buffer A is now easier to see.



The graph now

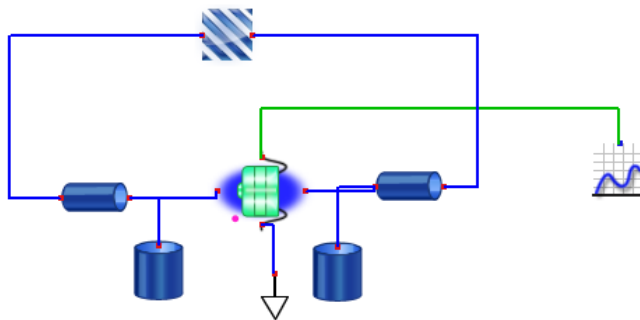


Example 2: Microphone directivity

The *plane wave source* in the *Acoustical* menu is used to create two pressure points that differ in phase according to a plane wave propagating in an infinite space. The two connector nodes on the *plane wave source* can be thought of as two microphone ports on an infinite surface. The parameters for this source include the angle and distance separating these two microphone ports.

This example shows how Ares can be used to model a directional microphone and compute its directivity.

Step 1: Create the following model, using the *plane wave source*, *directional microphone*, *circular ports* and *volumes* from the *Acoustical* menu.



For this example, the ports have diameters of 1mm, and lengths of 2 mm. The volumes have diameters of 2 mm and heights of 1 mm. The default values for the directional microphone were used.

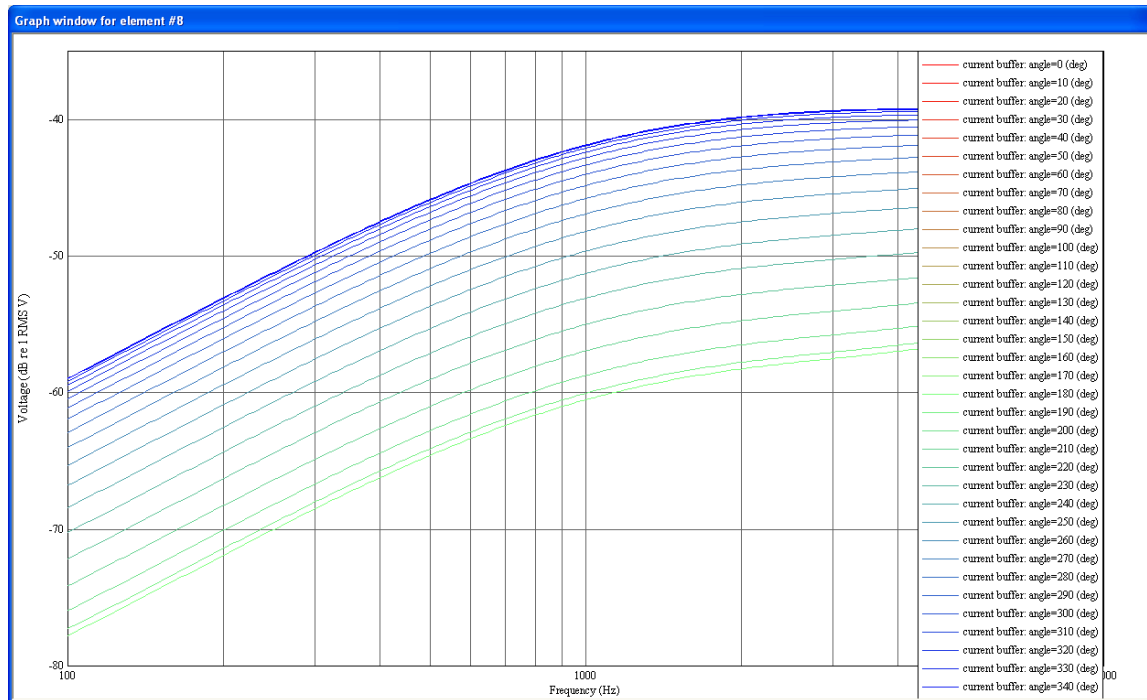
Step 2: Note that the default angle parameter for the plane wave source is 360 degrees. We're going to change the iteration range to be from 0 to 100%. By iterating on the 360 degree angle, this will cause the plane wave source to vary its angle from 0 to 360 degrees.

Element Params		Variables		Global Params	
description					
P	94.0	SPL	...		
angle	360.0	deg	...		
Dsep	7.0	mm	...		

Select the Global Params tab, and change the iteration parameters to min of 0%, max of 100%, and the number of iterations to 37. This will cause the angle to change by 10 degree steps.

Iteration parameter range	
min iteration	0 %
max iteration	100.0 %
num	37

Step 3: Press the “...” iterate button for the plane wave source’s angle parameter. The results are shown in the following plot. However, it’s difficult to see any directivity results from the microphone design.



Step 4: To see directivity, click on the graph icon and change the following plotting options:

- change type to “linear”
- check the *vs iterator @ Hz* option and enter the following frequencies to the right of the check box: 100 400 1000 2000 5000

quantity: voltage

type: linear

☒ vs iterator @Hz 100 400 1000 2000 5000

☐ normalize to y= 0 at x= 0

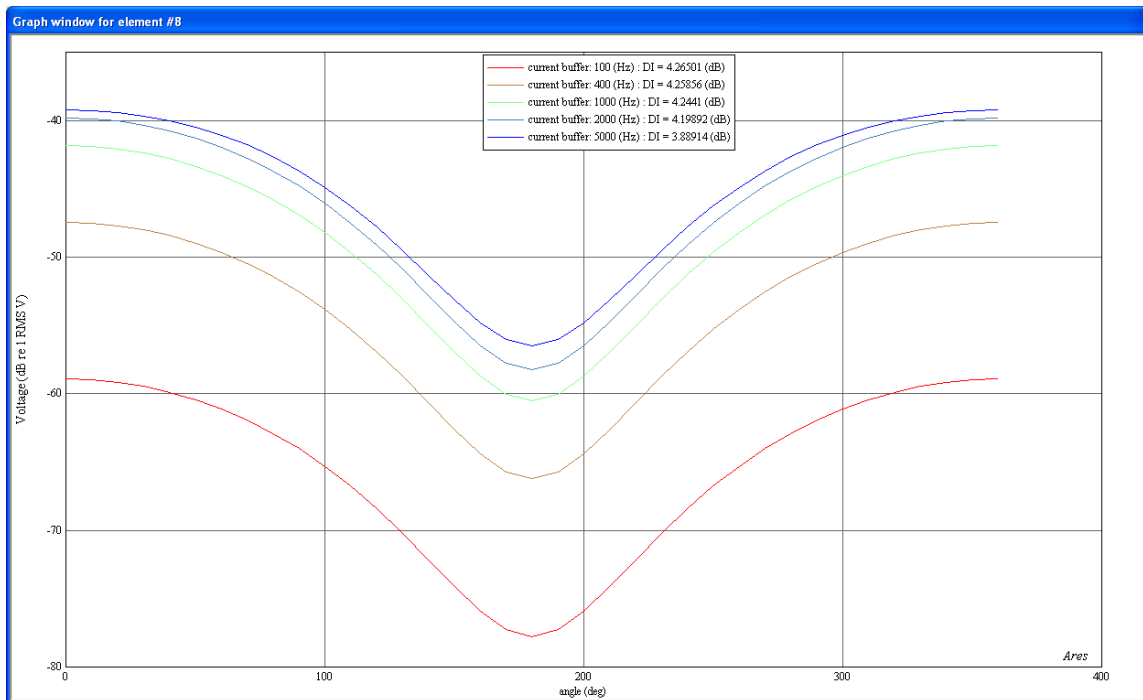
x axis range (min):

y axis range (min):

Update graph Hide graph

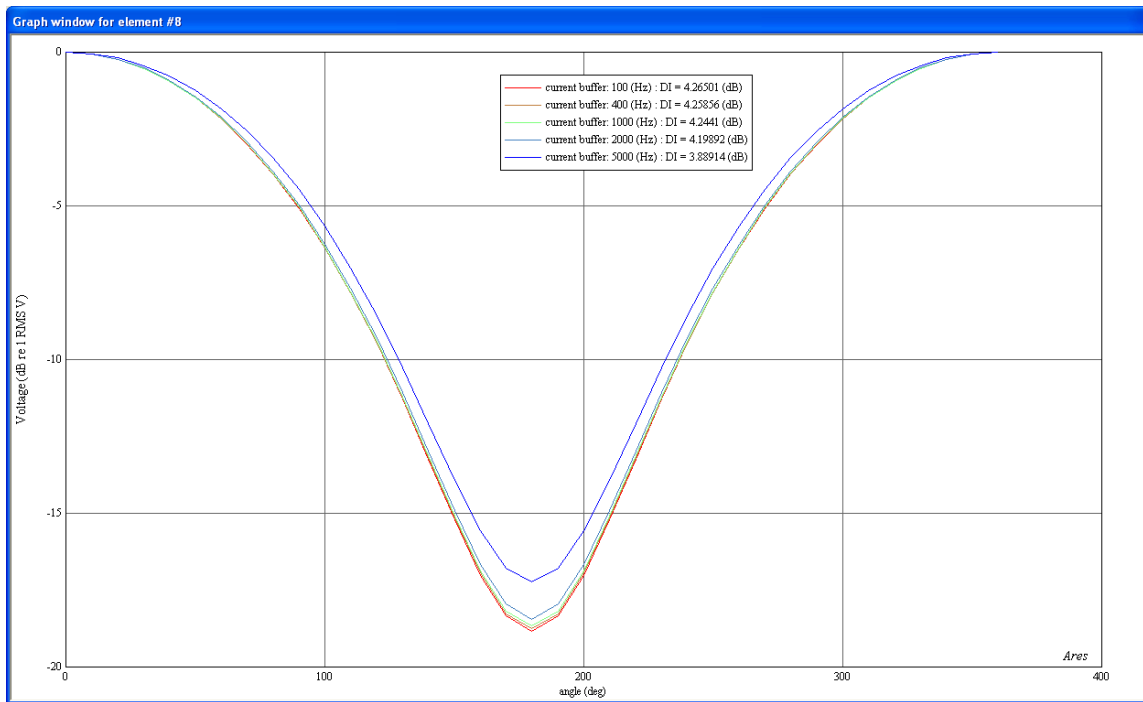
Set buffer line Advanced Options

Now look at the graph again. The response vs angle is now plotted as shown below. Note that Ares automatically computes the directivity (DI) for the graph and displays the result in the legend.



Note that all of the curves look very similar, but have an offset due to the sensitivity at each frequency. To remove this sensitivity offset for each curve, check the *normalize to y=* option in the graph window to produce the following plot.

☒ vs iterator @Hz 100 400 1000 20
☒ normalize to y= 0 at x= 0



While not as attractive as a polar plot, this does provide sufficient information to see the desired directivity for a microphone design.

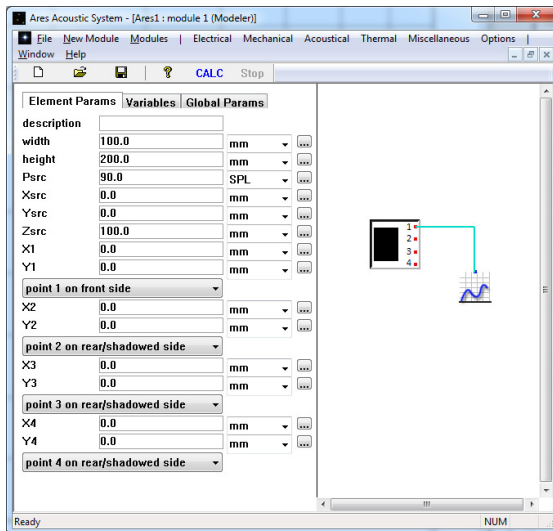
Example 3: Rigid Rectangle Model of Microphone on Tablet

This example shows how to use the rigid rectangle to predict the sound pressure on a tablet being used in a speaker phone application. The tablet is modeled as an infinitely thin rigid rectangle. The default 100mm wide by 200mm high size for the rectangle is used, which roughly corresponds to a 7 inch tablet.

Create a new Modeler module from the *New Module* menu.

Pull the *Acoustical* menu down and select *rigid rectangle* about one third way down the menu and place it in the Ares window.

Create a graph element from the Miscellaneous menu and connect it to the first node on the rigid rectangle as shown below.



The rigid rectangle element doesn't require any additional elements to predict the sound pressure on the surface of the rectangle. Before we run the solver, let's make sure that the frequency range is from 50 to 10000 Hz. Select the *Global Params* tab and set the Fmin, Fmax and num points to those shown below.

Element Params		Variables		Global Params	
Frequency range for analysis					
Fmin	50.0	Hz			
Fmax	10000.0	Hz			
num points	200				
<input checked="" type="checkbox"/> logarithmic frequency distribution					

Reselect the *Element Params* tab so we can see the element parameters again.

By default the *rigid rectangle* element has a height and width the size of a 7 inch tablet. The four sensing points are all at the origins 0,0 with the first one on the front and the rest on the back.

Let's move the last two to 1 cm from the edge. Set the Y3 and X4 parameters to the values shown below so the third point will be 1 cm from the top and the fourth point will be 1 cm from the right edge.

X3	0.0	mm	...
Y3	90.0	mm	...
point 3 on rear/shadowed side			
X4	40.0	mm	...
Y4	0.0	mm	...
point 4 on rear/shadowed side			

Now run the solve by either pressing the CALC button or pressing F5. The solve will rapidly run through the low frequencies, but take a longer and longer time as the frequency increases.

Copy all for positions to the A-D graph buffers by

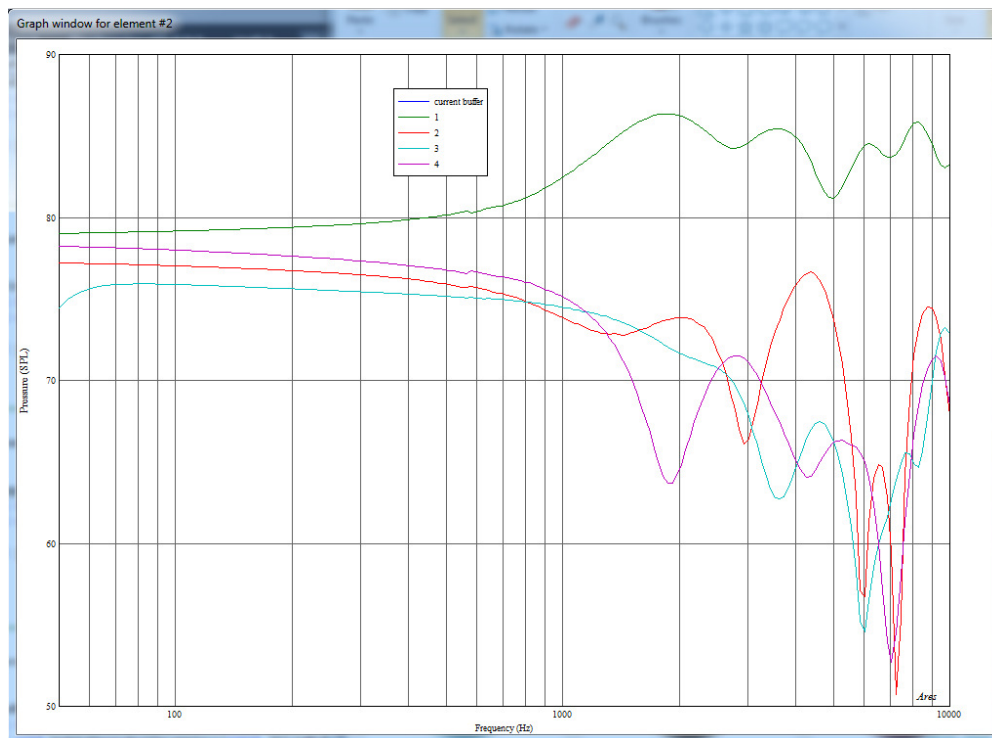
- connecting the graph to rigid rectangle's node 1
- press the A button on the graph controls to copy the node data into that buffer

- select the check box to the left of the A button
- type "1" into the comment for the A buffer

Repeat this for node 2, 3 and 4 so the graph element's parameters are as shown below.

Element Params	Variables	Global Params
description <input type="text"/>		
Connector node <input type="text"/>		
<input checked="" type="checkbox"/> <input type="text"/>		
Plotting buffers		
<input checked="" type="checkbox"/> A	1	<input type="text"/>
<input checked="" type="checkbox"/> B	2	<input type="text"/>
<input checked="" type="checkbox"/> C	3	<input type="text"/>
<input checked="" type="checkbox"/> D	4	<input type="text"/>
<input type="checkbox"/> E		<input type="text"/>

Finally, enter "50" for the *x axis range (min max)* field to force the graph to start at 50 Hz. The graph element's graph will now appear as shown below.



Note that the first point, which is the only one on the front of the rectangle, has the highest pressure which roughly has a 6 dB increase at higher frequencies.

Points 2,3, and 4 have their pressures drop off at higher frequencies as the rectangle is shadowing those sensing points.

At low frequencies, the curves are all approaching the same values which result from the wavelengths being so long that the rectangles size being too small to significantly effect the sound wave. Note that the 3rd point (light blue curve) is curving down as the frequency

approaches 50 Hz. This is an artifact of the approximate solution used and not physically meaningful.

The iteration feature can be used to show how the pressure changes across the back of the rectangle. To do this, we'll iterate across the y axis from the center of the rectangle at y=0 mm to the top at y=100mm.

First go to the *Global Params* tab and set the iteration parameters to those shown below.

Iteration parameter range		
min iteration	0	%
max iteration	100.0	%
num	10.0	

Going from 0% to 100% will cause the parameter to go from zero (0%) to its current value (100%).

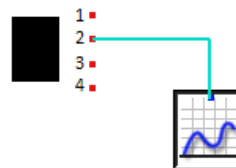
Now select *Element Params* tab, select the rigid rectangle element, and change the X2 and Y2 parameters to 0 100mm respectively.

X2	0.0	mm	...
Y2	100	mm	...
point 2 on rear/shadowed side			

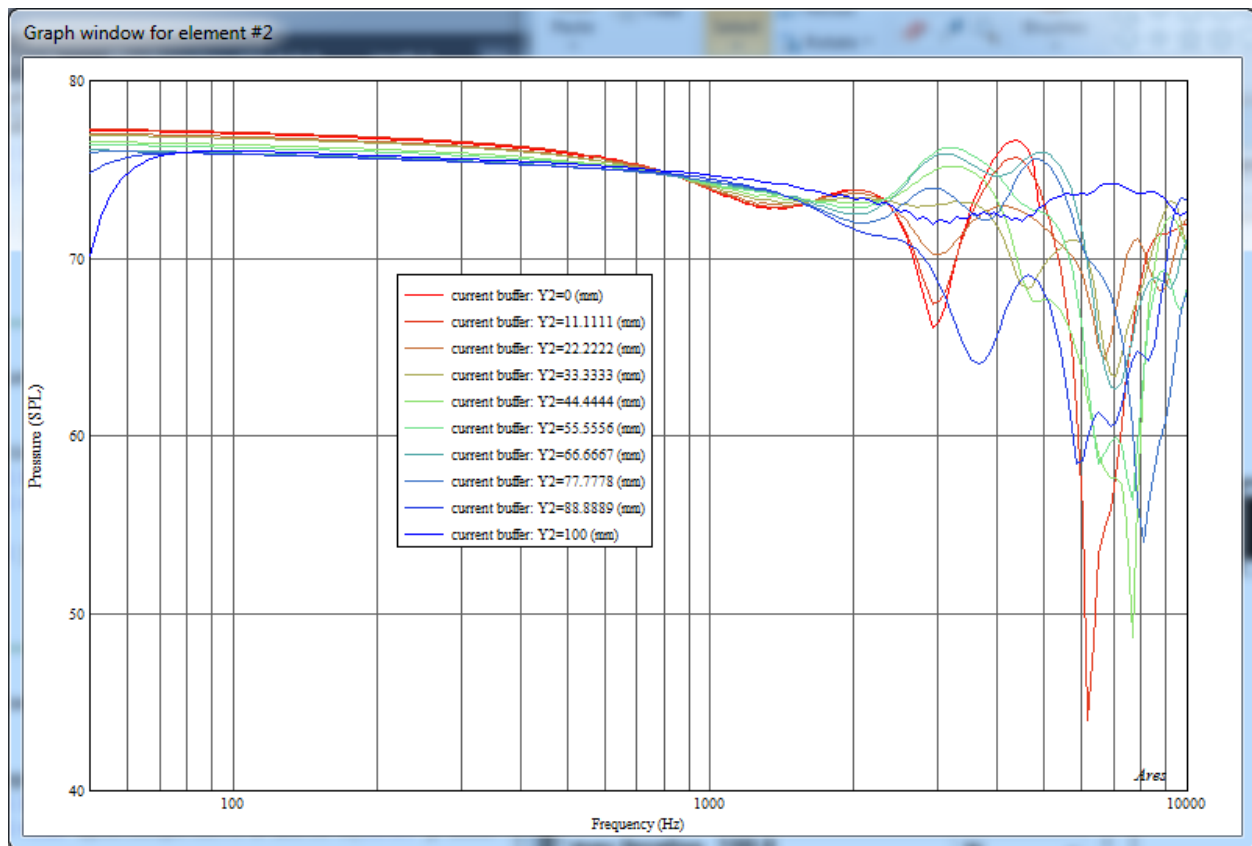
Now press the iterate button to the left of Y2. This will run the solve 11 times, which will take a while.

To plot the values for just the second point, connect the graph element to node 2 and uncheck buffers A-D for plotting as shown below.

Element Params	Variables	Global Params
description		
Connector node		
<input checked="" type="checkbox"/>		
Plotting buffers		
<input type="checkbox"/> A	1	
<input type="checkbox"/> B	2	
<input type="checkbox"/> C	3	
<input type="checkbox"/> D	4	
<input type="checkbox"/> E		
<input type="checkbox"/> F		



The result should be the graph shown below. Little effect at low frequencies, with a lot of activity at the high frequencies.



Cross sections of these iteration curves can be made by selecting the *vs iterator @ graph* feature. Enter frequencies of 1000 4000 and 8000 for this parameter. Change the graph type from log x to linear, and clear the *x axis range (min max)* entry as shown below.

quantity

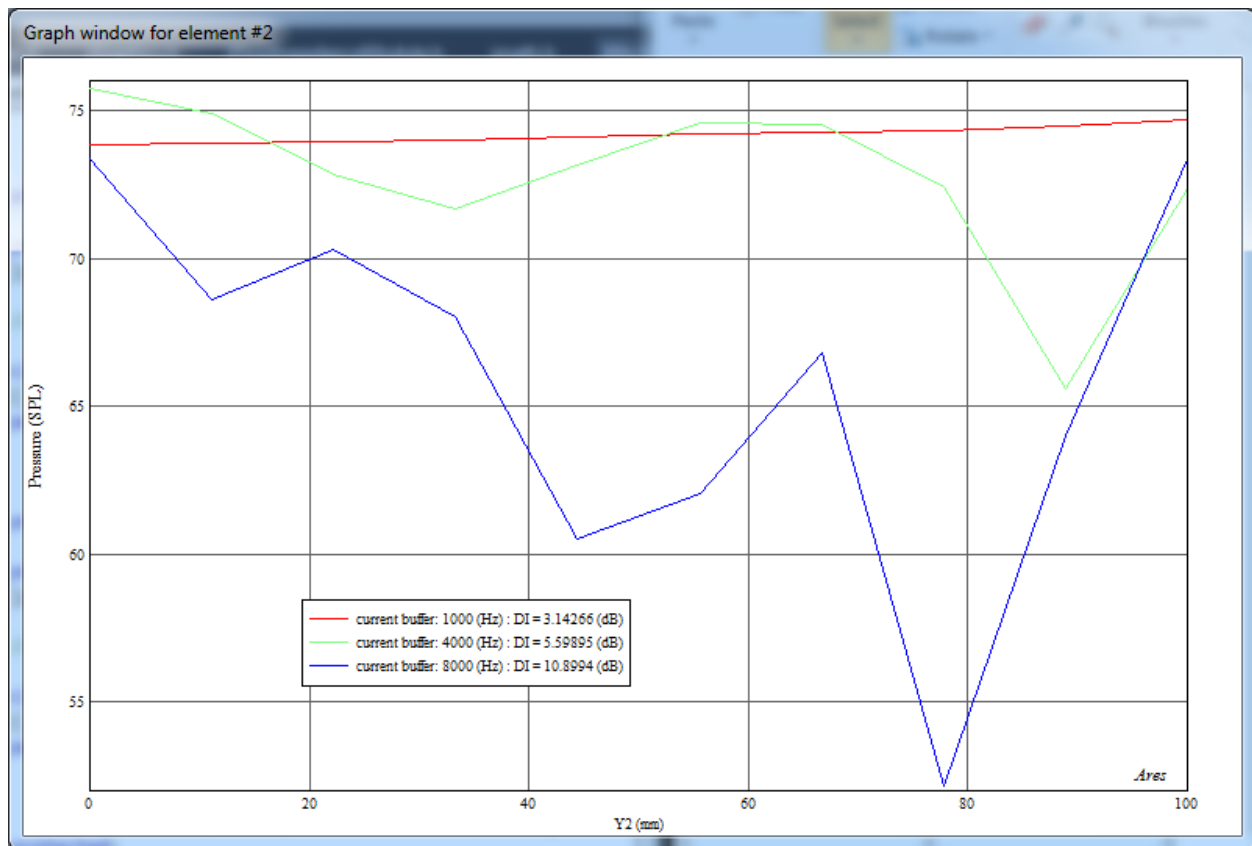
type

☒ vs iterator @Hz

☐ normalize to y= at x=

x axis range (min
y axis range (min

The result is the graph below which shows the pressure across the back of the rectangle, from the center at Y2=0 to the top at Y2=100mm for the three frequencies we entered.



There is little variation in the pressure across the back at 1 kHz, more at 4 kHz, and a great deal at 8 kHz, showing that the shadowing increases with increasing frequency.

For more information about Ares and acoustical measurement and modeling tools and services, contact MAE at:

info@MAELLC.COM
support@MAELLC.COM
(678) 234-5079

Or see us at MAELLC.COM.